

دانشگاه صنعتي امیرکبیر  
(پلی تکنیک تهران)

دانشكده مهندسی کامپیوتر

پایان‌نامه کارشناسی

سامانه نرم‌افزاری مقايسه الگوريتم‌هاي يافتن دو مسير‌ هميلتوني با بیشتر خوانایی در گراف هندسي كامل

نگارش

پارسا عبداللهی

استاد راهنما

دکتر علیرضا باقری

اسفند ۱۴۰۰

تکمیل آینده صفحه فرم ارزیابی و تصویب پایان نامه- فرم تأیید اعضاء كميته دفاع

در این صفحه (هر سه مقطع تحصيلي) بايد فرم ارزيابي یا تایید و تصویب پایان­نامه/رساله موسوم به فرم کمیته دفاع براي ارشد و دكترا و فرم تصويب براي كارشناسي، موجود در پرونده آموزشی را قرار دهند.

اينجانب پارسا عبداللهی متعهد مي‌شوم كه مطالب مندرج در اين پايان نامه حاصل كار پژوهشي اينجانب تحت نظارت و راهنمايي اساتيد دانشگاه صنعتي اميركبير بوده و به دستاوردهاي ديگران كه در اين پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذكر گرديده است. اين پایان نامه قبلاً براي احراز هيچ مدرك هم‌سطح يا بالاتر ارائه نگرديده است.

در صورت اثبات تخلف در هر زمان، مدرك تحصيلي صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پيگيري قانوني خواهد داشت.

كليه نتايج و حقوق حاصل از اين پایان نامه متعلق به دانشگاه صنعتي اميركبير مي‌باشد. هرگونه استفاده از نتايج علمي و عملي، واگذاري اطلاعات به ديگران يا چاپ و تكثير، نسخه‌برداري، ترجمه و اقتباس از اين پایان نامه بدون موافقت كتبي دانشگاه صنعتي اميركبير ممنوع است.   
نقل مطالب با ذكر مآخذ بلامانع است.

تکمیل آینده در صفحه تعهدنامه اصالت اثر، در قسمت بالا سمت چپ، تاریخ دفاع خود را جایگزین تاریخ نوشته شده کنید.

پارسا عبداللهی

امضا

**برای مادر و پدرم؛ (تکمیل آینده)**

**تشکر و قدردانی**

با تشکر از استاد گرانقدرم دکتر علیرضا باقری که همیشه پذیرا بودند و با راهنمایی‌های ارزشمندشان مرا در انجام این پروژه هدایت کردند.

# چكيده

تکمیل آینده

در این پژوهش، چهار الگوریتم‌ برای بهینه کردن زیرمسئله‌ای از این مسائل ارائه و با هم مقایسه شده‌اند. زیرمسئله مذکور، کشیدن دو مسیر همیلتونی با کمترین تقاطع با خود و یکدیگر بر روی یک گراف هندسی کامل است.

دو الگوریتم از این چهار الگوریتم ابداعی است. یکی از الگوریتم‌ها از مقالات استخراج شده و به منظور کمینه کردن تعداد تقاطع‌های یکی مسیر با خود استفاده می‌شود. الگوریتم چهارم نیز الگوریتم ژنتیک است.

تکمیل آینده نتایج

واژه‌های کلیدی:

یال، راس، مسیر تکمیل آینده

|  |  |
| --- | --- |
| فهرست مطالب | صفحه |

[چكيده ‌أ](#_Toc28770794)

[فصل اول مقدمه (دستور العمل)مقدمه 1](#_Toc28770795)

[فصل دوم مشخصات یک پایان نامه و گزارش علمی 11](#_Toc28770796)

[مشخصات یک پایان نامه و گزارش علمی 12](#_Toc28770797)

[2-1- برخورداری از غنای علمی 12](#_Toc28770798)

[2-2- ارجاع به‌موقع و صحیح به منابع دیگر 12](#_Toc28770799)

[2-3- ساده‌نویسی 12](#_Toc28770800)

[2-3-1- وحدت موضوع 13](#_Toc28770801)

[2-3-2- اختصار 13](#_Toc28770802)

[2-3-3- رعایت نكات دستوري و نشانه‌گذاري 13](#_Toc28770803)

[2-3-4- توجه به معلومات ذهنی مخاطب 14](#_Toc28770804)

[2-3-5- رعایت مراحل اصولی نگارش 14](#_Toc28770805)

[فصل سوم نگارش صحيحنگارش صحيح 15](#_Toc28770806)

[3-1- فارسي‌نويسي 16](#_Toc28770807)

[3-2- رعایت املاي صحيح فارسي 17](#_Toc28770808)

[3-3- رعایت قواعد نشانه‌گذاري 17](#_Toc28770809)

[3-3-1- ويرگول و نقطه 17](#_Toc28770810)

[3-3-2- دو نقطه 17](#_Toc28770811)

[3-3-3- گيومه 18](#_Toc28770812)

[3-3-4- نشانه پرسشی 18](#_Toc28770813)

[3-3-5- خط تیره 18](#_Toc28770814)

[3-3-6- پرانتز 18](#_Toc28770815)

[فصل چهارم سبك ها و قلم ها سبك ها و قلم ها 20](#_Toc28770816)

[4-1- قلم‌هاي فارسي 21](#_Toc28770817)

[4-2- قلم‌هاي انگلیسی 22](#_Toc28770818)

[4-3- فرمول‌ها(روابط رياضي) 23](#_Toc28770819)

[4-4- فاصله‌هاي افقي و عمودي 24](#_Toc28770820)

[4-4-1- فاصله كلي از چهار طرف كاغذ 24](#_Toc28770821)

[4-4-2- فاصله خط‌ها 24](#_Toc28770822)

[4-4-3- فاصله‌هاي تفكيك‌كننده 24](#_Toc28770823)

[4-5- فواصل بين كلمات 25](#_Toc28770824)

[4-6- جدانوشتن كلمات بدون گذاشتن فاصله بين آنها 25](#_Toc28770825)

[4-7- فهرست گزارش، فهرست شكل‌ها و فهرست جداول 25](#_Toc28770826)

[4-8- سربرگ و ته‌برگ (Header and Footer) 26](#_Toc28770827)

[4-9- جداول، منحني‌ها، شكل‌ها 26](#_Toc28770828)

[4-10- ارجاع به جداول، شكل‌ها، روابط، مراجع و بخش‌ها 27](#_Toc28770829)

[فصل پنجم بررسي ساختار پایان نامهبررسي ساختار پایان نامه 28](#_Toc28770830)

[5-1- بررسي سرفصل‌ها 29](#_Toc28770831)

[5-2- بررسي ساختار كلي 29](#_Toc28770832)

[5-3- بررسي مفهومي 29](#_Toc28770833)

[5-4- مطالعه مفهومي و جمله‌بندي 30](#_Toc28770834)

[5-5- تنظیم بندها 30](#_Toc28770835)

[5-6- بررسي قواعد نگارشي 31](#_Toc28770836)

[5-7- بررسي روابط 31](#_Toc28770837)

[5-8- بررسي شكل‌ها 32](#_Toc28770838)

[5-8-1- بررسي كيفيت شكل و تطابق عنوان آن 32](#_Toc28770839)

[5-8-2- بررسي تطابق روابط، برنامه و شكل 32](#_Toc28770840)

[5-9- بررسي جداول 32](#_Toc28770841)

[5-9-1- بررسي كيفيت جدول و تطابق عنوان آن 32](#_Toc28770842)

[5-9-2- بررسي تطابق روابط، برنامه و جدول 33](#_Toc28770843)

[5-10- به‌روز‌رساني مراجع 33](#_Toc28770844)

[5-11- صفحه‌بندي 33](#_Toc28770845)

[5-12- سربرگ و ته‌برگ‌ها 33](#_Toc28770846)

[فصل ششم جمع‌بندي و نتيجه‌گيري و پیشنهاداتجمع‌بندي و نتيجه‌گيري 35](#_Toc28770847)

[منابع و مراجع 37](#_Toc28770848)

[4-5- یا مطابق دستور العمل زیر : 38](#_Toc28770849)

[پيوست‌ها 47](#_Toc28770850)

[Abstract 48](#_Toc28770851)

|  |  |
| --- | --- |
| فهرست اشكال | صفحه |

[شكل ‏1‌-‌‌4- فرايند كواكستروژن . 26](#_Toc276969410)

|  |  |
| --- | --- |
| فهرست جداول | صفحه |

[جدول 4-1- قلم‌هاي فارسي 21](#_Toc276969411)

[جدول ‏2‌-4- قلم‌هاي انگلیسی. 22](#_Toc276969412)

[جدول ‏3-4- قلم و سبك فرمول‌ها. 23](#_Toc276969413)

[جدول ‏4‌-‌4- اندازه فرمول‌ها. 24](#_Toc276969414)

[جدول ‏5-‌4- عنوان جدول. 26](#_Toc276969415)

|  |
| --- |
| فهرست علائم |

علائم لاتين

|  |  |  |
| --- | --- | --- |
|  |  | ارتفاع |
|  |  | طول موج توربولانس |
|  |  | پريود توربولانس |
|  |  | سرعت تعادل وسيله پرنده |
|  |  | مولفه سرعت تندباد در راستاي محور طولي دستگاه مختصات بدني نسبت به اينرسي |

علائم يوناني

|  |  |  |
| --- | --- | --- |
|  |  | چگالي طيفي قدرت توربولانس |
|  |  | شدت توربولانس |
|  |  | بسامد توربولانس |
|  |  | بسامد فاصله‌اي |
|  |  |  |

بالا‌نويس‌ها

|  |  |  |
| --- | --- | --- |
|  |  | دستگاه مختصات بدني |

زيرنويس‌ها

|  |  |  |
| --- | --- | --- |
|  |  | تندباد (گاست) |

فصل اول  
مقدمه

با پیشرفت تکنولوژي در عصر حاضر، استفاده از محتواهای گرافیکی برای نمایش نقشه‌های مختلف بیشتر شده است. با توجه به اینکه بسیاری از این نقشه‌ها برای انتقال مفهوم به عموم استفاده می‌شود، - مثل نقشه ایستگاه‌های مترو شهری – خوانایی نقشه‌ها بیش از پیش اهمیت پیدا می‌کند. یکی از روش‌های مهم برای ترسیم نقشه‌ها، استفاده از مدل‌های گراف است. در این پایان‌نامه حالت خاصی از این مسئله به کمک الگوریتم‌های مختلف و سامانه گرافیکی طراحی شده بررسی شده‌ است.

۱.۱ صورت مسئله

گراف هندسی G = (S, E)، شامل مجموعه‌ رئوس G بر روی صفحه مختصات و مجوعه‌ یال‌ E که پاره‌خط مستقیم میان دو نقطه از G هستند، است. در مسئله این پایان‌نامه باید برای رئوس این گراف، دو مسیر همیلتونی رسم کنیم؛ به طوری که خوانایی این دو مسیر بیشینه مقدار ممکن باشد.

برای خوانایی گراف می‌توان به طرق مختلفی تعریف ارائه کرد. با توجه به اینکه یکی از کاربردهای این جور مسائل برای کشیدن نقشه‌ها می‌باشد، خوانایی باید با توجه به بینایی انسان تعریف شود. با توجه به اینکه تعریف دقیق این معیار کار بسیار پیچیده‌ای است و بسته به انسان‌های مختلف تعاریف مختلفی وجود دارد، به منظور ساده‌سازی، در این پایان‌نامه از سه معیار تقاطع دو مسیر با یکدیگر، تقاطع یک مسیر با خود و تعداد یال‌های مشترک بین دو مسیر برای تقریب میزان **ناخوانایی** استفاده شده است. با توجه به اینکه تاثیر این سه معیار در ناخوانا کردن نقشه یکسان نیست، در حالت‌های مختلف وزن‌های متفاوتی به این سه معیار داده شده است تا بتوان خوانایی نقشه را در چند حالت تعریف کرد. لازم به ذکر است با توجه به اینکه یال مشترک میان دو مسیر و تقاطع یک مسیر با خودش، نسبت به تقاطع دو مسیر با یکدیگر، تاثیر بیشتری بر کاهش خوانایی نقشه دارند، این پایان‌نامه نیز بر روی کاهش این دو معیار تاکید بیشتری دارد.

۲.۱ کارهای پیشین تکمیل آینده

در گذشته کارهای متفاوتی در این حوزه انجام شده است. در بسیاری از مقالات تمرکز بر پوشش گراف کامل هندسی با استفاده از زیرگراف‌های مسطح (مثل درخت‌های پوشا، تطابق‌های کامل یا مسیرهای همیلتونی) است. بعضی از مقالات نیز روی کم کردن تقاطع‌های بین این زیرگراف‌ها کار کرده‌اند. (تکمیل آینده: اشاره به مقاله؟! بیشتر بگم؟! پیدا کردن مقاله برای تقاطع‌ها؟!)

مسئله «کشیدن گراف‌ها» (graph drawing) پیرامون موضوع درست کردن نمایش‌های زیبا و کاربردی از گراف‌ها مطرح می‌شود. یکی از مهم‌ترین کاربرد‌های این مسئله، نمایش گراف‌ها به طوری است که این نمایش برای ذهن انسان قابل درک باشد و بتواند ارتباط اشیا داخل گراف را به خوبی منتقل کند. یک گراف می‌تواند مدلی برای بسیاری از ساختارها باشد و نمایش مناسب و قابل درک اجزای این ساختار نیاز به کشیده شدن مناسب آن گراف دارد.

یکی از کاربرد‌های روزمره و ساده این مسئله، نمایش ایستگاه‌های مترو شهری به طوری است که برای بیننده به راحتی قابل درک باشد و ارتباط ایستگاه‌های شهری در آن مشخص باشد.

برای واضح بودن یک گراف کشیده شده، معیارهای مختلفی تعریف شده است. یکی از این معیار‌ها، تعداد تقاطع‌های گراف (crossing) است. گرافی که تقاطع نداشته باشد، گراف مسطح نامیده می‌شود. در این پروژه، ما نیز این معیار را به عنوان معیار اصلی مقایسه الگوریتم‌ها قرار داده‌ایم و تلاش میکنیم تا الگوریتم‌هایی ارائه دهیم که تعداد تقاطع‌ها را کمینه کنند.

یکی از مسائل معروف پیرامون مسئله «کشیدن گراف‌ها»، مسئله «جایگذاری همزمان گراف‌ها» (simultaneous graph embedding) است. در این مسئله، تعدادی گراف مسطح n-راسی داریم و میخواهیم آن‌ها را به نقاط دلخواهی در صفحه متناظر کنیم، جوری که نمایش هر گراف با استفاده از آن n نقطه در صفحه، مسطح باشد. دقت کنیم که در این تناظر، اگر برای مثال راس v\_1 از گراف G\_1 را به نقطه p\_i تناظر دادیم، راس v\_1 از هر گراف دیگری را نیز باید به نقطه p\_i تناظر دهیم. مقالات مختلفی انواع و اقسام این مسئله را توضیح داده‌اند. [1]

در مسئله «جایگذاری همزمان گراف‌ها»، هر یال میتواند یک خم جردن بین دو نقطه در فضای مختصات باشد. حال اگر در مسئله جایگذاری گراف‌ها، مکان هندسی نقاط از قبل مشخص باشد، می‌توان الگوریتم‌های متفاوتی برای حالات خاصی از این مسئله مطرح کرد. در [2] تکنیکی ارائه می‌شود که اگر تناظر رئوس یک گراف مسطح به نقاط در صفحه مشخص باشد، می‌توان گراف را طوری کشید که هر یال، به طور میانگین حداکثر 16/3n – 1 پیچ (bend) داشته باشد.

حال اگر این شرط را بگذاریم که یال بین دو نقطه حتما باید پاره‌خط بین آن دو باشد، مسئله ما به مسئله «جایگذاری همزمان گراف‌های هندسی» (simultaneous embedding on geometric graphs) تبدیل می‌شود. به گراف‌هایی که یال‌های آن‌ها، دقیقا پاره‌خط بین نقاط متناظر رئوس آن در صفحه مختصات دکارتی باشند، گراف‌های هندسی گفته می‌شود.

یک edge packing، مجموعه‌ای از چندین زیرگراف یال‌مجزا از یک گراف است. یک edge partitioning یک edge packing است که تمامی یال‌های گراف را می‌پوشاند. در [2] ثابت می‌شود که مسئله edge partitioning یک مسئله NP-Complete می‌باشد. با این وجود، مطرح کردن این مسئله روی گراف‌های خاص، می‌تواند مفید باشد. در [3] و [4] بر روی حداکثر تعداد تطابق‌های بیشینه موجود در یک گراف کامل محدب کار شده است. در [5] نیز حد پایینی برای تعداد دور‌های همیلتونی موجود در یک گراف مسطح آورده شده است.

یک دور همیلتونی، دور است که شامل همه رئوس گراف بشود. پیدا کردن دور همیلتونی در یک گراف، حتی با فرض مسطح بودن آن، NP-Complete است [6]. از آنجایی که پیدا کردن بیش از ۱ دور همیلتونی مسطح و یال‌مجزا در یک گراف هندسی لزوما ممکن نیست، مقالاتی روی این مسئله کار کرده‌اند که اگر حداکثر اجازه ۱ تقاطع را به هر دور بدهیم، حداکثر چند دور همیلتونی یال‌مجزا می‌توان پیدا کرد. همچنین در [7] ثابت می‌شود حداقل k-1 دور همیلتونی، با ویژگی‌هایی که وصف شد، در یک گراف هندسی کامل که در آن هیچ سه نقطه‌ای هم‌خط نیستند وجود دارد به طوریکه n = 2k + hکه h عددی کمتر از 2k  می‌باشد. همچنین الگوریتمی برای پیدا کردن این دورها ارائه داده می‌شود. با توجه به اینکه پیدا کردن دور‌های همیلتونی سختتر از پیدا کردن مسیر‌های همیلتونی است، مقاله فوق ثابت میکند که اگر هیچ سه نقطه‌ای هم خط نباشند، میتوان حداقل k مسیر همیلتونی غیر خود متقاطع و یال مجزا پیدا کرد. البته تفاوت مسئله مورد بررسی در [7] با مسئله مورد بررسی در این پروژه این است که شرطی درباره هم‌خط نبودن هیچ سه‌نقطه‌ای وضع نشده است و امکان دارد حتی همه نقاط هم‌خط باشند (که البته در این حالت، دو مسیر همیلتونی یال مجزا وجود ندارد). غیر از الگوریتم یاد شده، الگوریتم دیگری برای مسائل مشابه یافت نشد.

نکته قابل توجه در اکثر مقالات بررسی شده، این پیش‌فرض است که نقاط تا جای ممکن همخط نیستند؛ به طوری که در بعضی از مقالات فرض شده‌است هیچ سه‌ نقطه داده شده بر روی یک خط نمی‌باشند. در این پایان‌نامه پیش‌فرضی بر روی نحوه قرارگیری نقاط در صفحه مختصات نداریم.

۳.۱ ساختار پایان‌نامه

در فصل دوم ابتدا چهار الگوریتم استفاده شده به همراه کلیت پیاده‌سازی آن‌ها شرح داده ‌می‌شود. همچنین راجع به ابعاد مختلف آن‌ها مثل مرتبه زمانی و مقدار حافظه مورد نیازشان بررسی‌هایی صورت خواهد گرفت. (تکمیل آینده: بیشتز از این نوع‌آوری‌هامو بگم؟!)

سپس معماری و نحوه پیاده‌سازی سامانه نرم‌افزاری شرح داده خواهد شد. در این سامانه کاربر می‌تواند به صورت دستی نقاطی را در صفحه مختصات مشخص کند و به دلخواه خود به هر کدام از سه معیار سنجش خوانایی وزن دهد. سپس به صورت گرافیکی خروجی الگوریتم‌ها به نمایش در می‌آید و کاربر می‌تواند خوانایی خروجی الگوریتم را در عمل بسنجد.

در فصل سوم، نتیجه آزمایش‌ حالت‌های مختلف گراف‌های ورودی بر روی الگوریتم‌ها آورده شده. همچنین با توجه عملکرد الگوریتم‌ها را مقایسه و سپس نتیجه‌گیری می‌کنیم. در انتها نیز راجع به کارهای آتی صحبت می‌شود.

در فصل آخر نیز مراجع و فریم‌ورک‌های استفاده شده آورده شده‌است.

۴.۱ تعاریف و مفاهیم

۱.۴.۱ گراف کامل (Complete Graph)

گراف کامل [گراف ساده‌ای](https://fa.wikipedia.org/wiki/%DA%AF%D8%B1%D8%A7%D9%81_(%D8%B3%D8%A7%D8%AE%D8%AA%D8%A7%D8%B1_%D8%AF%D8%A7%D8%AF%D9%87)) است که در آن هر راس، دقیقا یک یال به هر کدام از رئوس دیگر دارد. درواقع، گراف کامل راسی، شامل یال مختلف می‌شود.

۲.۴.۱ گراف هندسی (Geometric Graph)

گراف هندسی G = (S, E)، شامل مجموعه‌ رئوس G بر روی صفحه مختصات و مجموعه‌ پاره‌خط مستقیم میان دو نقطه از G هستند، است.

۳.۴.۱ مسیر همیلتونی

مسیر همیلتونی مسیری در یک [گراف ساده](https://fa.wikipedia.org/w/index.php?title=%DA%AF%D8%B1%D8%A7%D9%81_%D8%B3%D8%A7%D8%AF%D9%87&action=edit&redlink=1) است که هر راس را دقیقاً یک بار مشاهده می‌کند.

۴.۴.۱ پوش محدب (Convex hull)

پوشش محدب مجموعه از نقاط در صفحه مختصات است؛ به طوری که کوچکترین مجموعه محدبی است که شامل مجموعه نقاطی خاص و از قبل انتخاب شده می‌باشد.

۵.۴.۱ گراف مسطح (Planar Graph)

گراف مسطح به گرافی گفته می‌شود که بتوان یال‌های آن را طوری کشید که به جز در رئوس گراف، نقطه تماس دیگری نداشته باشند.

۶.۴.۱ گراف مسطح هندسی

گراف مسطح هندسی، گرافی مسطح است که یال‌های آن از کشیدن پاره‌خط‌ها مستقیم بین نقاط ایجاد شده‌اند.

۷.۴.۱ الگوریتم ژنتیک

الگوریتم‌ ژنتیک، یک تکنیک جستجو برای یافتن راه‌حل تقریبی برای بهینه‌سازی یک مدل ریاضی است. الگوریتم ژنتیک نوع خاصی از الگوریتم‌های تکاملی است که از تکنیک‌های [زیست‌شناسی فرگشتی](https://fa.wikipedia.org/wiki/%D8%B2%DB%8C%D8%B3%D8%AA%E2%80%8C%D8%B4%D9%86%D8%A7%D8%B3%DB%8C_%D9%81%D8%B1%DA%AF%D8%B4%D8%AA%DB%8C) مانند وراثت، جهش و اصول انتخابی داروین برای یافتن حالت بهینه استفاده می‌کند.

۵.۱ ابزارها و کتابخانه‌های پیاده‌سازی

پیاده‌سازی الگوریتم‌ها با زبان سی‌پلاس‌پلاس (C++) انجام شده است. در پیاده‌سازی الگوریتم ژنتیک نیز از کتابخانه OpenGA استفاده شده است.

سامانه نرم‌افزاری، به کمک فریم‌ورک جنگو (Django) پیاده‌سازی شده‌است. برای قسمت بک‌اند (Backend) از زبان برنامه‌نویسی پایتون (Python) و برای قسمت فرانت‌اند از اچ‌تی‌ام‌ال (HTML)، سی‌اس‌اس (CSS) و جاوا اسکریپت (JavaScript) استفاده شده‌است. همچنین از کتابخانه PaperJS برای پیاده‌سازی صفحه مختصات بهره گرفته شده‌است.

فصل دوم  
الگوریتم‌های استفاده شده

در این فصل به معرفی و بررسی الگوریتم‌های استفاده شده در این پایان‌نامه می‌پردازیم. ابتدا الگوریتم محاسبه معیار ناخوانایی برای دو مسیر همیلتونی را توضیح می‌دهیم و سپس به سراغ الگوریتم‌ها می‌رویم.

دو الگوریتم اول ابداعی بوده و سومی از مقالات این حوزه استخراج شده است. الگوریتم‌های ابداعی، الگوریتم‌هایی تقریبی (approximation) بوده و داخل یک چتر تئوری، به دنبال بهترین حالت می‌گردند. الگوریتم سوم قطعی (exact) بوده و تعداد تقاطع‌های یک مسیر با خودش و یال‌های مشترک میان دو مسیر را تقریبا به صفر می‌رساند. الگوریتم چهارم نیز الگوریتم‌ ژنتیک است که بر خلاف سه الگوریتم دیگر، در دسته الگوریتم‌های حوزه هوش مصنوعی قرار می‌گیرد.

۱.۲ الگوریتم محاسبه ناخوانایی دو مسیر همیلتونی

۱.۱.۲ شرح الگوریتم

ابتدا الگوریتم باید سه معیار تقاطع دو مسیر با یکدیگر، تقاطع یک مسیر با خود و تعداد یال‌های مشترک را محاسبه کند. برای محاسبه تقاطع دو مسیر با یکدیگر، برخورد هر یال از مسیر همیلتونی اول را با هر یک از یال‌های مسیر همیلتونی دوم چک می‌کنیم. برای محاسبه تقاطع یک مسیر با خود، برخورد هر یال از یک مسیر را با دیگر یال‌های آن بررسی می‌کنیم. برای محاسبه یال‌های مشترک بین دو مسیر نیز هر کدام از یال‌های مسیر اول را با هر یک از یال‌های پاره‌خط دوم مقایسه می‌کنیم.

در مرحله آخر، با توجه به وزن هر کدام از این سه معیار، مقدار ناخوانایی نهایی محاسبه و برگردانده می‌شود.

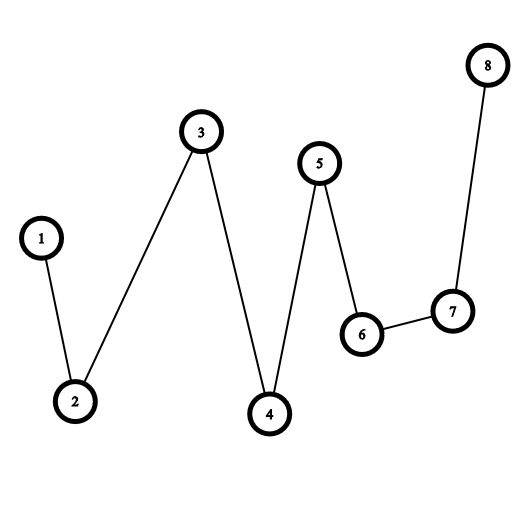
۲.۲.۲ تحلیل مرتبه زمانی و حافظه

با توجه به مقایسه دوبه‌دوی یال‌ها، مرتبه زمانی الگوریتم از خواهد بود. لازم به ذکر است حافظه نیز از خواهد بود.

۲.۲ الگوریتم «چرخش نقاط»

۱.۲.۲ شرح الگوریتم

اگر نقاط را بر اساس x-شان مرتب کنیم و بر همین اساس هم یک مسیر همیلتونی بکشیم، این مسیر حتما خودش را قطع نخواهد کرد؛ چرا که برای ایجاد تقاطع نیاز به کم شدن x مسیر، حداقل در بعضی قسمت‌های آن است درحالیکه x مسیر ایجاد شده هیچ‌گاه کم نمی‌شود.



شکل 1: این مسیر خودش را قطع نمیکند

این ویژگی فقط مربوط به محور x نبوده و اگر نسبت به هر محور دیگری نقاط را مرتب کنیم، رخ خواهد داد. به عبارت دیگر، اگر صفحه را به میزان دلخواهی بچرخانیم و بعد نقاط را بر اساس x جدیدشان مرتب کرده و مسیر را بکشیم، باز هم مسیر هیچ تقاطعی با خودش نخواهد داشت.

در الگوریتم چرخش نقاط، ابتدا صفحه را به اندازه‌ای تصادفی می‌چرخانیم و بر اساس ترتیب x نقاط جدید، مسیر همیلتونی اول را رسم می‌کنیم. سپس همین کار را تکرار می‌کنیم تا مسیر دوم بدست آید. سپس میزان ناخوانایی را برای این دو مسیر بدست می‌آوریم و با جواب بهینه‌ای که الگوریتم از قبل پیدا کرده مقایسه می‌کنیم. همین روش به تعداد خیلی زیادی تکرار می‌شود و در نهایت بهترین جواب پیدا شده برگردانده می‌شود.

همانطور که از برمی‌آید، این الگوریتم تقاطع یک مسیر با خودش را به صفر می‌رساند اما این کار به هزینه افزایش تعداد تقاطع‌های دو مسیر با هم و تعداد یال‌های مشترک انجام می‌شود.

۱.۲.۲ تحلیل مرتبه زمانی و حافظه

مرتبه زمانی چرخش صفحه به میزانی تصادفی و مرتب سازی نقاط نیز است. در نتیجه مرتبه زمانی یکبار محاسبه دو مسیر خواهد بود. با توجه به اینکه بعد از محاسبه مسیر‌ها باید الگوریتم محاسبه ناخوانایی اجرا شود، هزینه کل یکبار اجرای این روش خواهد شد. نکته مهم این است طبق آزمایش‌های انجام شده، برای گرفتن جواب مطلوب، الگوریتم باید بیش از ده هزار بار اجرا شده و زاویه چرخش صفحه مختصات تا حداقل ۴ رقم بعد از اعشار انتخاب شود. از این رو در عمل زمان اجرای این الگوریتم کندتر از آنچه از مرتبه‌ زمانی‌اش می‌آید است.

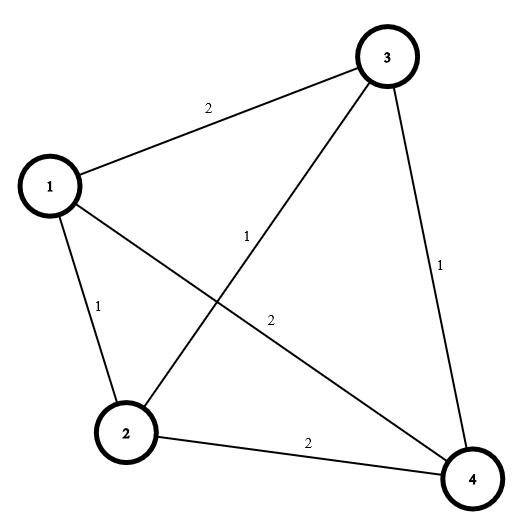
لازم به ذکر است مرتبه حافظه نیز خواهد بود.

۳.۲ الگوریتم «جایگشت چهارتایی»

۱.۳.۲ شرح الگوریتم

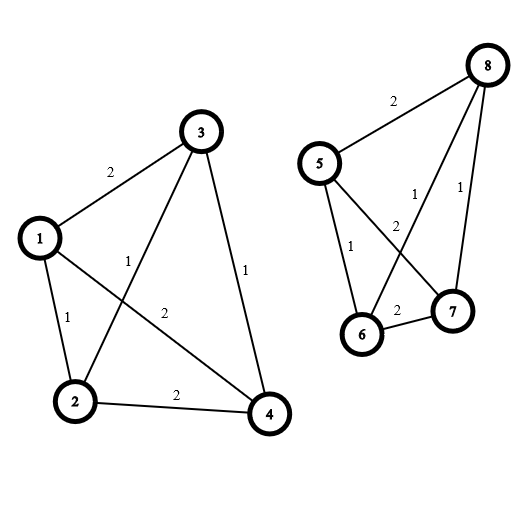
در این الگوریتم ابتدا نقاط از بر اساس مقدار مختصات x-شان از چپ به راست مرتب و شماره‌گذاری می‌شوند. سپس به دسته‌های چهارتایی تقسیم می‌شوند.

می‌دانیم گراف کامل چهار راسی، شش یال دارد و هر مسیر همیلتونی که این چهار راس را دربربگیرد باید از سه تا از این شش یال استفاده کند. همچنین ۲۴ (تعداد چینش‌های مختلف اعداد یک تا چهار در یک سطر) مسیر همیلتونی مختلف، به ازای چهار راس، وجود دارد. در این روش ابتدا یکی از ۲۴ مسیر مختلف انتخاب می‌شود تا در مسیر همیلتونی اول قرار بگیرد. سپس سه یال باقی‌مانده از شش یال اولیه، در مسیر همیلتونی دوم قرار می‌گیرند (شکل ۱).



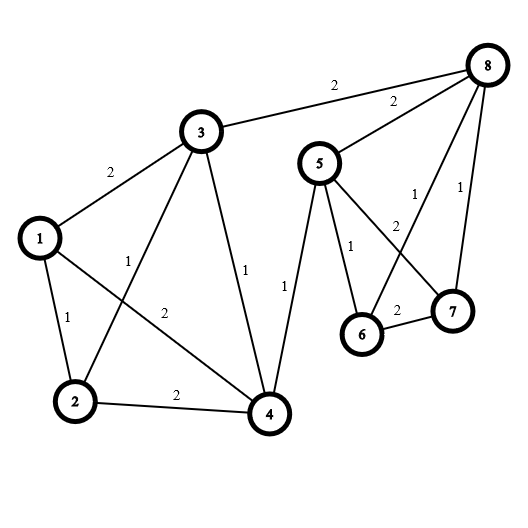
شکل 2 - یال‌های مسیر اول با عدد یک و مسیر دوم با عدد دو مشخص شده‌اند.

سپس، برای دسته دوم نقاط نیز یکی از ۲۴ حالت انتخاب می‌شود و دو مسیر همیلتونی برای دسته دوم نقاط کشیده می‌شود. مسیر همیلتونی شماره یک در دسته اول، دو سر دارد (رئوس یک و چهار). در دسته دوم نیز رئوس پنج و هشت دو سر مسیر همیلتونی شماره دو هستند (تصویر ۲).



شکل 3 – همانطور که ذکر شد، نقاط از چپ به راست مرتب شده‌اند.

حال با اتصال یکی از رئوس یک و چهار با یکی از رئوس پنج و هشت، مسیر همیلتونی شماره یک در دسته اول و دوم به هم متصل خواهند شد. به طرز مشابه، همین اتفاق برای مسیر همیلتونی شماره دو رخ می‌دهد (شکل ۳).



شکل 4 - مسیر همیلتونی شماره یک و دو برای ۸ راس اول کشیده شده است.

به همین ترتیب تمامی دسته‌ها به هم متصل می‌شوند تا در نهایت دو مسیر همیلتونی ساخته شود. سپس با امتحان همه ۲۴ حالت کشیدن مسیر‌های داخل یک دسته و همه حالت‌های اتصال دو دسته متوالی به یکدیگر، بهترین جواب بدست آمده (با توجه به معیار ناخوانایی)، از میان همه حالت‌ها انتخاب و به عنوان جواب نهایی الگوریتم برگردانده می‌شود.

حال امکان دارد تعداد نقاط ورودی بر چهار بخش‌پذیر نباشد. در این حالت می‌توان حداکثر سه نقطه انتها را از الگوریتم حذف کرد و برای بقیه نقاط مسئله را حل کرد. در انتها نیز این نقاط را به ترتیب از چپ به راست به انتهای هر دو مسیر اضافه می‌کنیم. البته این نحوه اضافه کردن قسمت آخر هر دو مسیر را ناخواناتر می‌کند اما باید دقت کرد که این اتفاق حداکثر برای سه نقطه رخ می‌دهد.

۲.۳.۲ تحلیل مرتبه زمانی و حافظه

اگر فرض کنیم نقطه روی صفحه مختصات داریم، این الگوریتم تمام ۲۴ حالت یک دسته و ۱۶ حالت اتصال دو دسته به هم را امتحان می‌کند. با حذف حالت‌های تکراری در پیاده‌سازی، می‌توان یک ضریب اعمال کرد. با در نظر گرفتن هزینه الگوریتم محاسبه ناخوانایی، مرتبه زمانی نهایی کد به حالت زیر خواهد شد:

با توجه به نمایی بودن مرتبه زمانی، امکان اجرای این الگوریتم برای تعداد رئوس بالا وجود ندارد. به عبارت دقیق‌تر، یک کامپیوتر معمولی می‌تواند حداکثر تا ۱۵ نقطه ورودی را در زمان کمتر از ۱۰ ثانیه اجرا کرده و خروجی را برگرداند. لازم به ذکر است مرتبه حافظه نیز خواهد بود.

تکمیل آینده: سودوکد بیارم؟

## ۴.۲ الگوریتم «زیگزاگ» (تکمیل آینده: منبع)

۱.۳.۲ شرح الگوریتم

این الگوریتم روی کمینه کردن تعداد یال‌های مشترک و تقاطع یک مسیر با خودش تمرکز می‌کند. از این رو الگوریتم زیگ‌زاگ برای حالت‌هایی که ناخوانایی روی برخورد دو مسیر با هم تاکید بیشتری دارد مناسب نخواهد بود.

برای شروع الگوریتم، ابتدا یکی از رئوسی که روی پوش محدب همه رئوس قرار دارد انتخاب می‌شود. آن را راس p می‌نامیم. برای ساده‌سازی، در پیاده‌سازی الگوریتم نقطه‌ای که کمترین مقدار Y را دارد انتخاب شده است.

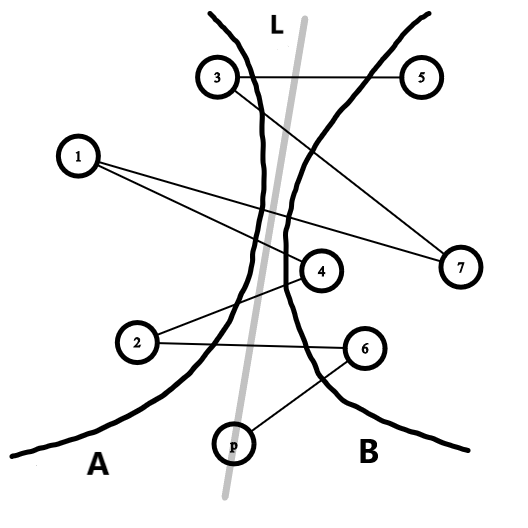
با توجه به اینکه p از همه راس‌ها پایین‌تر است، امکان ندارد روی پوش محدب نباشد چرا که اگر داخل پوش محدب باشد، حتما ضلعی از پوش وجود دارد که نقطه‌ای از آن پایین‌تر از p است. با توجه به اینکه در یک پوش محدب، حتما حداقل یکی از رئوس کمترین مقدار Y را میان همه نقاط رو یا داخل پوش دارد، نتیجه می‌شود که یکی از رئوس پوش محدب Y اکیدا کمتری نسبت به p دارد و این تناقض است.

بعد از پیدا کردن p ، بقیه رئوس را از دیدگاه p به صورت ساعت‌گرد مرتب و نصف کرده و به دو گروه تقسیم می‌کنیم. این گروه‌ها را A و B می‌نامیم. همچنین می‌توان خطی از p کشید که رئوس A و B را از هم جدا کند. این خط را L می‌نامیم. بدون کم شدن از صورت مسئله، فرض می‌کنیم .

حال دو مسیر میسازیم.

۱.۱.۳.۲ مسیر همیلتونی اول تکمیل آینده: منبع

مسیر اول، که آن را زیگ‌زاگ می‌نامیم، مسیری است که به طور یک در میان بین رئوس مجموعه A و B رفت و آمد می‌کند؛ به عبارت دیگر، تمامی یال‌های آن L را قطع می‌کنند. برای ساخت این مسیر، پوش محدب رئوس داخل A و B که هنوز به مسیر اضافه نشده‌اند را در نظر می‌گیریم. حال و را در نظر می‌گیریم به طوری که p1q1 ضلع سمت چپ پوش محدب باشد. حال q1 را به مسیر اضافه کرده و از B حذف می‌کنیم. سپس مجددا همین کار را تکرار می‌کنیم اما این بار p1 را به مسیر اضافه می‌کنیم. در واقع، هر دفعه، بعد از پیدا کردن p1 و q1، اگر راس قبلی‌ای که به مسیر اضافه کردیم از مجموعه B بود، p1 را و اگر از مجموعه A بود q1 را به انتهای مسیر همیلتونی اضافه کرده و از مجموعه رئوس باقی‌مانده حذف می‌کنیم. این روش را تا انتها ادامه می‌دهیم و در نهایت مسیر همیلتونی اول شکل می‌گیرد (شکل ۲).



شکل 5: مسیر همیلتونی اول

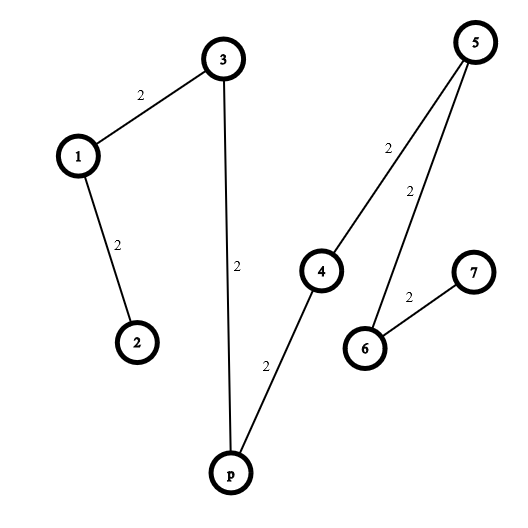
در این روش ساخت مسیر، با توجه به اینکه در هر لحظه، مسیر ساخته شده تقاطعی با پوش محدب نقاط خارج از مسیر ندارد (البته با فرض هم‌خط نبودن هیچ سه نقطه‌ای) و نقطه جدیدی که اضافه خواهد شد نیز این ویژگی را حفظ می‌کند، می‌توان نتیجه گرفت که این مسیر هیچگاه خودش را قطع نخواهد کرد.

همچنین، با اضافه شدن یال جدید به مسیر، فاصله نقطه برخورد خط L و یال جدید از راس p همیشه بیشتر می‌شود (البته با فرض هم‌خط نبود هیچ سه نقطه‌ای) و این با خودمتقاطع بودن مسیر تناقض ایجاد می‌کند. پس مسیر نمی‌تواند خودش را قطع کند. البته قابل توجه است که اگر حداقل سه راس روی صفحه هم‌خط باشند، امکان دارد یال‌های زیگ‌زاگی روی هم بیوفتند که در این صورت تقاطع رخ داده‌است.

۲.۱.۳.۲ مسیر همیلتونی دوم تکمیل آینده: منبع

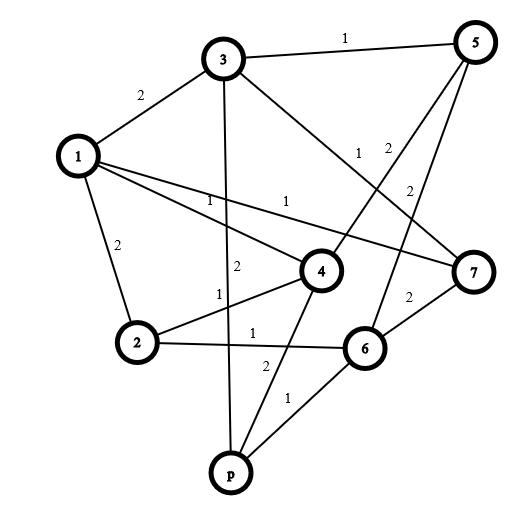
مسیر دوم از اتصال دو زیر مسیر ساخته می‌شود که هر کدام اعضای داخلی مجموعه‌های A و ‌B را به هم متصل می‌کنند. ابتدا رئوس داخل مجموعه A را از نگاه p، به صورت ساعت‌گرد مرتب کرده و به همان ترتیب به هم متصل می‌کنیم. با توجه به نقاط مرتب شده هستند، این زیرمسیر با خودش تقاطعی نخواهد داشت. سپس انتهای زیرمسیر را به p وصل می‌کنیم. با توجه به اینکه نقطه انتهایی زیرمسیر A، از نظر ساعت‌گردی آخرین نقطه مجموعه A می‌باشد، یال بین این راس و p، با دیگر یال‌های زیرمسیر برخوردی نخواهد داشت. به این ترتیب زیرمسیر A ساخته می‌شود.

به طور مشابه، نقاط مجموعه B را نیز به هم وصل می‌کنیم تا زیرمسیر B ایجاد شود. با توجه به اینکه خط L بین رئوس مجموعه‌های A و B فاصله انداخته است، زیرمسیر A و B با هم تقاطعی ندارند. حال p را به نقطه اول زیرمسیر B یا نقطه آخر زیر مسیر B وصل می‌کنیم. در هر صورت، این یال با زیر مسیر A و B تقاطعی نخواهد داشت اما امکان دارد یکی از دو یال را در مسیر زیگ‌زاگ استفاده کرده باشیم. با توجه به اینکه در مسیر زیگ‌زاگ دقیقا یک یال از p به مجموعه B وجود دارد، p می‌تواند حداقل به یکی از رئوس اول یا آخر مجموعه B متصل شود. این یال را نیز اضافه می‌کنیم و مسیر دوم ساخته می‌شود (شکل ۶).



شکل 6: مسیر همیلتونی دوم

با توجه به اینکه زیرمسیر‌های A و B داخل مجموعه‌های A و B را به هم وصل می‌کنند اما مسیر همیلتونی اول (مسیر زیگ‌زاگ) یک در میان بین دو مجموعه در رفت و آمد بود، می‌توان نتیجه گرفت که یال مشترکی میان دو مسیر وجود نخواهد داشت (شکل ۷).



شکل 7: یال‌های مسیر اول با ۱ و مسیر دوم با ۲ مشخص شده‌اند

۲.۴.۲ تحلیل مرتبه زمانی و حافظه

الگوریتم پیدا کردن پوش محدب از مرتبه زمانی است. با توجه به اینکه در محاسبه مسیر زیگ‌زاگ، به ازای اضافه کردن یک راس به مسیر نیاز به گرفتن یک پوش محدب از نقاط باقی‌مانده داریم، پس دقیقا N-1 باز پوش محدب محاسبه خواهد شد. در نهایت، مرتبه زمانی یافتن مسیر زیگ‌زاگ است.

پیدا کردن مسیر دوم نیز تنها نیاز به مرتب سازی ساعت‌گرد نقاط دارد و از مرتبه زمانی است. پس در نهایت، مرتبه زمانی الگوریتم، خواهد بود. لازم به ذکر است مرتبه حافظه نیز خواهد بود.

۵.۲ الگوریتم ژنتیک

در این قسمت، به منظور رعایت اختصار مطلب، از توضیح الگوریتم ژنتیک خودداری می‌کنیم و به تنظیم پارامتر‌ها و توابع پیاده‌سازی شده به منظور گرفتن بهترین خروجی از این الگوریتم می‌پردازیم. لازم به ذکر است که در توضیحات آینده، منظور از ژن، دو مسیر همیلتونی است.

۱.۵.۲ تابع محاسبه برازش یا تناسب (fitness)

تابع محاسبه تناسب الگوریتم، همان تابع بدست آوردن میزان ناخوانایی دو مسیر همیلتونی است که در قسمت ۱.۲ توضیح داده شد.

۲.۵.۲ توابع انتخاب والد

برای انتخاب والد، دو تابع پیاده‌سازی شده است. تابع اول، بر اساس مقدار تناسب به یک نسل وزن می‌دهد؛ به طوری که هر چه مقدار کمتر باشد، احتمال انتخاب توسط الگوریتم بیشتر است.

دومین تابع، تابع rank selection است. این تابع بر اساس مقدار تناسب، تمامی اعضای یک نسل را مرتب کرده و بر اساس جایگاه‌شان در آرایه مرتب شده به آن‌ها وزن می‌دهد؛ به طوری که هر چه جایگاه والد بهتر باشد، شانس بیشتری برای انتخاب شدن دارد. در بررسی‌های انجام شده، این تابع عملکرد بهتری داشته است و در نتیجه به عنوان تابع انتخاب والد مورد استفاده قرار می‌گیرد.

۳.۵.۲ توابع جهش

برای جهش، دو تابع پیاده‌سازی شد. تابع اول، ابتدا عددی یک رقمی را به صورت تصادفی انتخاب می‌کند و سپس به همان تعداد کار پیش رو را انجام می‌دهد. ابتدا به صورت تصادفی یکی از دو مسیر ژن را انتخاب می‌کند و سپس به صورت تصادفی جای دو راس متوالی در آن را با هم عوض می‌کند. با این کار مسیر کمی متفاوت می‌شود و در برخی موارد، تقاطع‌ها نیز اصلاح خواهند شد.

در تابع دوم نیز ابتدا عددی یک رقمی به صورت تصادفی انتخاب می‌شود و سپس به همان تعداد کار پیش رو انجام می‌شود. ابتدا به صورت تصادفی یکی از دو مسیر ژن انتخاب شده و سپس به صورت تصادفی جای دو راس (نه لزوما متوالی) در آن با هم عوض می‌شود. با همین کار ساده، امکان دارد مسیر خیلی متفاوت شود. در بررسی‌های انجام شده، استفاده از تابع دوم منجر به این شد که الگوریتم ژنتیک خیلی دیرتر در بهینه محلی گیر کند و در نتیجه خروجی بهتر و بهتر شود که این باعث بهتر شدن جدی خروجی نهایی الگوریتم ژنتیک شد. البته این روش نسبت به روش قبلی کمی کندتر است و بهتر شدن با سرعت کمتری اتفاق می‌افتد اما در مجموع تابع دوم برای انجام جهش‌ها انتخاب شد.

۴.۵.۲ توابع تقاطع (crossover)

برای این قسمت، سه تابع پیاده‌سازی شد. تابع اول، تابعی بسیار ساده است که برای بدست آوردن یک ژن از دو ژن والد، از هر کدام یکی از مسیر‌ها را به صورت تصادفی انتخاب می‌کند و به این شکل ژن جدید به دست می‌آید. مشکل این روش این بود اولا منجر به ایجاد مسیر‌های همیلتونی جدید نمی‌شد و از این نظر زیادی به جهش متکی می‌شدیم. همچنین این روش خیلی زودتر به بهینه‌های محلی میل می‌کند چرا که اگر دو والد خیلی از هم دور باشند، خروجی تقاطعشان نیز به احتمال زیاد تناسب خیلی بدی خواهد داشت و هر چه والدین به هم نزدیک‌تر باشند، تناسب تقاطعشان بهتر خواهد شد.

در هر کدام از دو تابع دوم و سوم، ابتدا دو به دو مسیر‌های ژن والد اول و دوم را به صورت رندم با هم جفت شده و تشکیل دو جفت می‌دهند؛ یعنی در هر جفت، یک مسیر از ژن اول و یک مسیر از ژن دوم وجود دارد. سپس از هر جفت، به صورتی که در ادامه خواهد آمد یک مسیر جدید بیرون می‌آید و در ژن بچه قرار داده می‌شود.

در تابع دوم که PMX Crossover نام دارد، ابتدا یک بازه L و R انتخاب می‌شود و همه این بازه در مسیر اول در مسیر نهایی قرار می‌گیرد. سپس روی بازه [L, R] مسیر دوم حرکت می‌شود و اگر راس دیده شده در همین بازه در مسیر اول بود، از آن می‌گذرد ولی اگر نبود، به کمک گراف جایگشتی که از روی مسیر اول و دوم ساخته شده، جای مناسب آن در مسیر نهایی را پیدا می‌کند. دقت شود که هر مسیر همیلتونی، در واقع یک جایگشت از شماره رئوس گراف است. {تکمیل آینده: منبع برای آینده بدم؟!}

در تابع سوم، به طور مشابه ابتدا بازه [L, R] مسیر اول در مسیر نهایی قرار می‌گیرد و آن‌ها را علامت‌گذاری می‌کنیم. سپس از ابتدا تا انتهای مسیر دوم را پیمایش می‌کنیم و اگر راس که از آن می‌گذریم قبلا علامت‌گذاری نشده بود، آن را در مسیر نهایی قرار می‌دهیم. در بررسی‌های انجام شده، این تابع بهتر از تابع دوم عمل کرد و منجر به خروجی‌های بهتری شد. از این رو این تابع به عنوان تابع تقاطع مورد استفاده قرار گرفت.

۵.۵.۲ پارامترهای الگوریتم

بر اساس آزمایش‌های مختلف، و با در نظر گرفتن دو معیار تناسب و مدت زمان اجرای الگوریتم، پارامتر‌های زیر برای اجرای الگوریتم ژنتیک انتخاب شد:

* با توجه به مدت زمان اجرای الگوریتم تعداد جمعیت هر نسل ۳۰۰ و تعداد حداکثر نسل تولید شده ۵۰۰ انتخاب شد.
* برای رسیدن به جواب بهینه، درصدی که از عمل تقاطع به نسل جدید اضافه می‌شوند شصت درصد انتخاب شد؛‌ یعنی چهل درصد از نسل قبلی مستقیما به نسل بعدی منتقل می‌شوند. همچنین احتمال جهش یک ژن از نسل جدید ۰.۴ انتخاب شد. احتمال کمتر باعث کندی الگوریتم و احتمال جهش بیشتر منجر به پرش زیاد الگوریتم می‌شد.

تکمیل آینده: تحلیل زمانی ژنتیک

فصل دوم  
سامانه نرم‌افزاری مقایسه الگوریتم‌ها

در این فصل به بررسی سامانه نرم‌افزاری ایجاد شده می‌پردازیم. ابتدا نیازمندی سامانه را توضیح می‌دهیم. سپس به متدولوژی توسعه و معماری آن را شرح می‌دهیم و در نهایت نحوه استفاده از سامانه را به کمک شکل توصیف می‌کنیم.

۱.۲ نیازمندی

به صورت کلی مسئله اصلی این پایان‌نامه درباره کشیدن خوانای گراف‌هاست و خوانایی معیاری است که از شخص به شخص می‌تواند متفاوت باشد؛ یعنی امکان دارد نقشه‌ای برای فردی کاملا گویا باشد به طوری که علائم و معانی داخل آن را به سرعت یاد بگیرد اما برای شخص دیگری کاملا گنگ و پیچیده به نظر بیاید. به همین منظور و برای تست واقعی الگوریتم‌های مختلف، نیاز به سامانه‌ای نرم‌افزاری احساس می‌شد تا کاربر بتواند در عمل کارایی الگوریتم‌های مختلف را بسنجد. همچنین برای تعریف بهتر معیار ناخوانایی که معیار ارزیابی الگوریتم‌ها است، نیاز بود تا در عمل تاثیر تعریف‌های مختلف سنجیده بشه.

۲.۲ متدلوژی توسعه

در سطح دنیا برای توسعه نرم‌افزارهای مختلف، متدولوژی‌های مختلفی از جمله متدولوژي چابک، آبشاری و هیبریدی مورد استفاده قرار می‌گیرد. در این پایان‌نامه، با توجه به اندازه پروژه و مشخص بودن تصویر کلی خروجی کار، متدولوژی آبشاری به عنوان روش توسعه نرم‌افزار انتخاب شد.

۱.۲.۲ متدولوژي توسعه آبشاری تکمیل آینده: منبع

متدولوژی آبشاری، روشی برای اجرای پروژه است که در آن، مراحل انجام کار، از پیش تعیین شده‌اند و به ترتیب انجام می‌شوند و زمانی که هریک از مراحل یا فازهای پروژه به اتمام رسید، مرحله بعد آغاز می‌شود. همان طور که این فرآیند به ترتیب و پیوسته است، زمانی که هر مرحله تکمیل شد امکان بازگشت به آن مرحله تقریباً وجود ندارد و هزینه بر است؛ بنابراین باید تا جای ممکن طبق برنامه اجرایی که اول کار چیده می‌شود عمل کرد (شکل ۷).



شکل 8: متدولوژی توسعه نرم‌افزار آبشاری

## **۲.۲ توسعه سامانه** تکمیل آینده: بازه زمانی هر کدووم از مراحلو بگم یا قسمت پیاده‌سازی یا verification یا maintenance رو بگم؟!

۱.۲.۲ نیازمندی‌ها

برای توسعه این نرم‌افزار، ابتدا نیازمندی‌های انجام پروژه در آمد. این نیازمندی‌ها به صورت کلی شامل قسمت‌های زیر شد:

* طراحی یا پیدا کردن الگوریتم‌های مناسب برای حل مسئله
* پیاده‌سازی الگوریتم‌ها به طوری که زمان اجرایی مناسبی داشته باشند تا نه تنها بتوان بر روی تعداد زیادی تست آن‌ها را اجرا کرد، بلکه کاربر بتواند به صورت برخط بهشان ورودی بدهد و خروجی را ببیند.
* پیاده‌سازی قسمتی برای ایجاد تست‌های مختلف تا الگوریتم‌ها را آزمایش و مقایسه کرد
* پیاده‌سازی قسمت بک‌اند پروژه به طوری که به راحتی بتواند با پیاده‌سازی الگوریتم‌ها و فرانت‌اند ارتباط برقرار کند و بر بستر وب باشد
* پیاده‌سازی قسمت فرانت‌اند به طوری که در عین سادگی پیاده‌سازی و زیبایی ظاهر بتواند نقطه ورودی بگیرد و مسیر‌های همیلتونی بدست آمده را نمایش دهد.

۲.۲.۲ طراحی

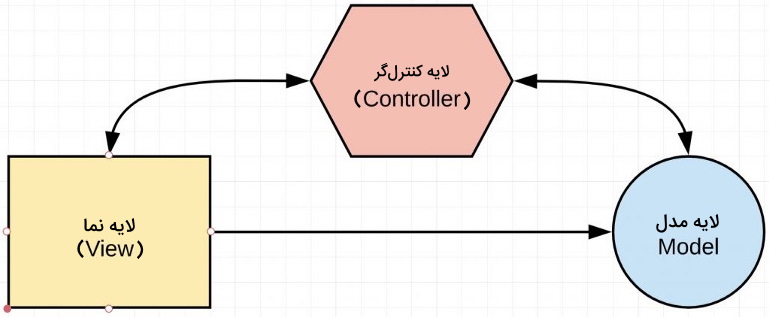
به منظور بالا رفتن سرعت اجرای الگوریتم‌ها و ساختن تست از زبان سی‌ پلاس پلاس (C++) استفاده شده‌است. همچنین از فریم‌ورک جنگو (که بر پایه مدل MVC است) به همراه زبان برنامه‌نویسی پایتون برای پیاده‌سازی قسمت بک‌اند استفاده شده است که هم راحت به فرانت‌اند متصل می‌شود و هم به خاطر کتاب‌خانه‌های قدرتمند پایتون می‌تواند کارهای دیگری از جمله اجرای الگوریتم‌ها را انجام دهد. برای قسمت فرانت‌اند، بعد از گشتن و امتحان کتاب‌خانه‌ها، تصمیم به استفاده از کتاب‌خانه PaperJS گرفته شد.

در ادامه ابتدا با معماری MVC آشنا شده و سپس معماری نرم‌افزار را تحلیل می‌کنیم.

۱.۲.۲.۲ معماری مدل، نما و کنترل‌گر (MVC) تکمیل آینده:‌ منبع بگم؟!

الگوی معماری MVC سرنامی برای عبارت Model-View-Controller است که در فارسی به معنای معماری مدل، نما و کنترل‌گر است. امروزه در اینترنت [وب‌سایت‌ها](https://blog.faradars.org/html-document-and-website-structure/) [تعاملی‌تر](https://blog.faradars.org/beginners-guide-to-interaction-design-1/) شده‌اند، از حالت ایستا درآمده‌اند و پویایی بیش‌تری دارند. الگوی معماری مدل، نما و کنترل‌گر، یکی از الگو‌های معماری مرسوم برای توسعه و ساخت اینجور برنامه‌ها است. در ابتدا، از این الگو برای رابط کاربری گرافیکی و برنامه‌های کامپیوتری استفاده می‌شد؛ اما امروزه، الگوی معماری MVC برای طراحی اپلیکیشن‌های مبتنی بر وب و [برنامه‌های کاربردی موبایل](https://blog.faradars.org/how-to-build-an-app/) به کار می‌رود.

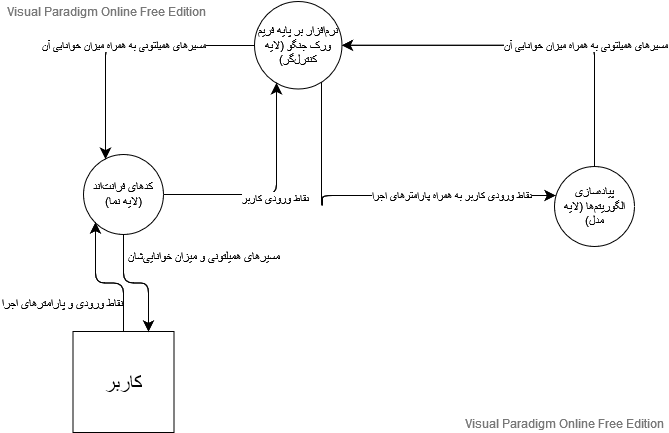
در الگوی معماری MVC یک سیستم نرم‌افزاری یا همان اپلیکیشن به سه قطعه تفکیک می‌‌شود. لایه نما (View) تنها می‌داند که چه طور باید [رابط کاربری](https://blog.faradars.org/9-simple-tips-to-improve-ui-designs/)‌ را نمایش دهد. اگر برای نمایش [رابط کاربری](https://blog.faradars.org/different-ways-to-approach-ui-design/) نیاز به هر نوع داده‌ای باشد، لایه نما آن را از لایه مدل دریافت خواهد کرد. اما نما مستقیماً از مدل برای پیدا کردن داده‌ها سوال نخواهد کرد؛ لایه نما این کار را از طریق لایه کنترل‌گر انجام می‌دهد. بنابراین، کنترل‌گر برای بدست آوردن داده‌های مورد نیاز نما، مدل را فراخوانی می‌کند. وقتی داده‌ها آماده شدند، کنترل‌گر لایه نما را از حاضر بودن داده‌ها با خبر می‌کند. اکنون، لایه نما می‌تواند داده‌ها را از مدل دریافت کند. این جریان انتقال داده را می‌توان به صورت زیر به تصویر کشید (شکل ۱۰)

[](https://blog.faradars.org/wp-content/uploads/2021/01/MVC-Architecture.jpg)

شکل ۹: توصیف مدل MVC

۲.۲.۲.۲ معماری سامانه نرم‌افزاری

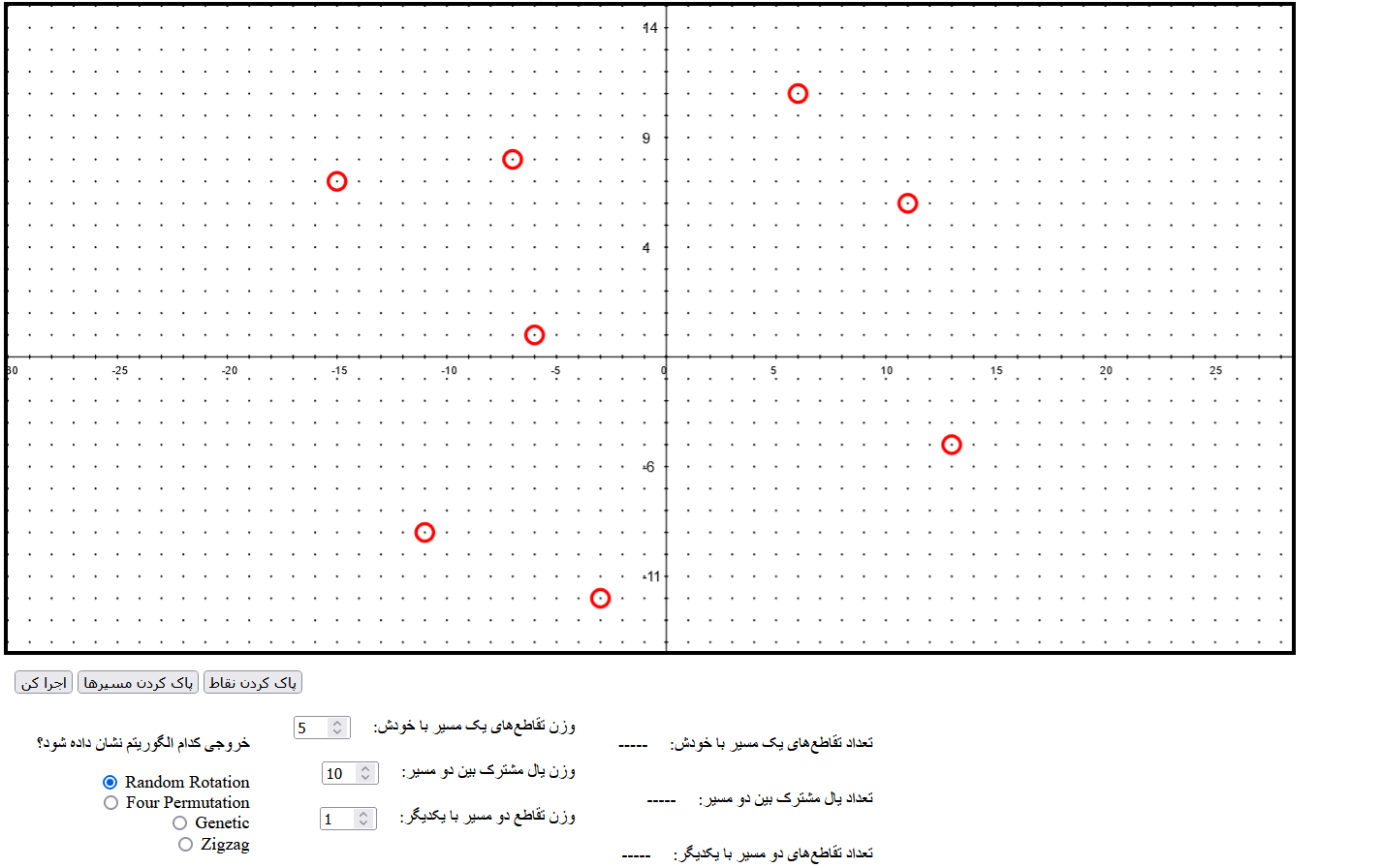
در سامانه پیاده‌سازی شده، لایه نما توسط HTML، CSS و JavaScript پیاده‌سازی شده‌است و زمانی که کاربر بخواهد نقاطی در صفحه مختصات اضافه کند، این نقاط به کمک کتابخانه PaperJS و زبان JavaScript اضافه می‌شوند. سپس زمانی که درخواست اجرای الگوریتم‌ها توسط کاربر داده شود، درخواستی از لایه نما به لایه کنترل‌گر که با زبان پایتون و در فریم‌ورک جنگو پیاده‌سازی شده‌اند، می‌رود. قسمت کنترل‌گر درخواست لایه نما را تبدیل به پارامترهای درستی برای اجرای لایه مدل می‌کند و سپس لایه مدل را اجرا می‌کند. لایه مدل الگوریتم‌های پیاده‌سازی شده با زبان C++ می‌باشند و بعد از اجرا، مسیر‌های همیلتونی را به لایه کنترل‌گر خروجی می‌دهند. سپس لایه کنترل‌گر چک‌های لازم برای معتبر بودن خروجی را انجام می‌دهد. بعد خروجی را تبدیل به فرمت مورد نیاز لایه نما کرده و به آن تحویل می‌دهد. در آخر لایه نما مسیر‌ها را به همراه میزان خوانایی‌شان بر روی صفحه مختصات به کاربر نمایش می‌دهد (شکل ۱۱).



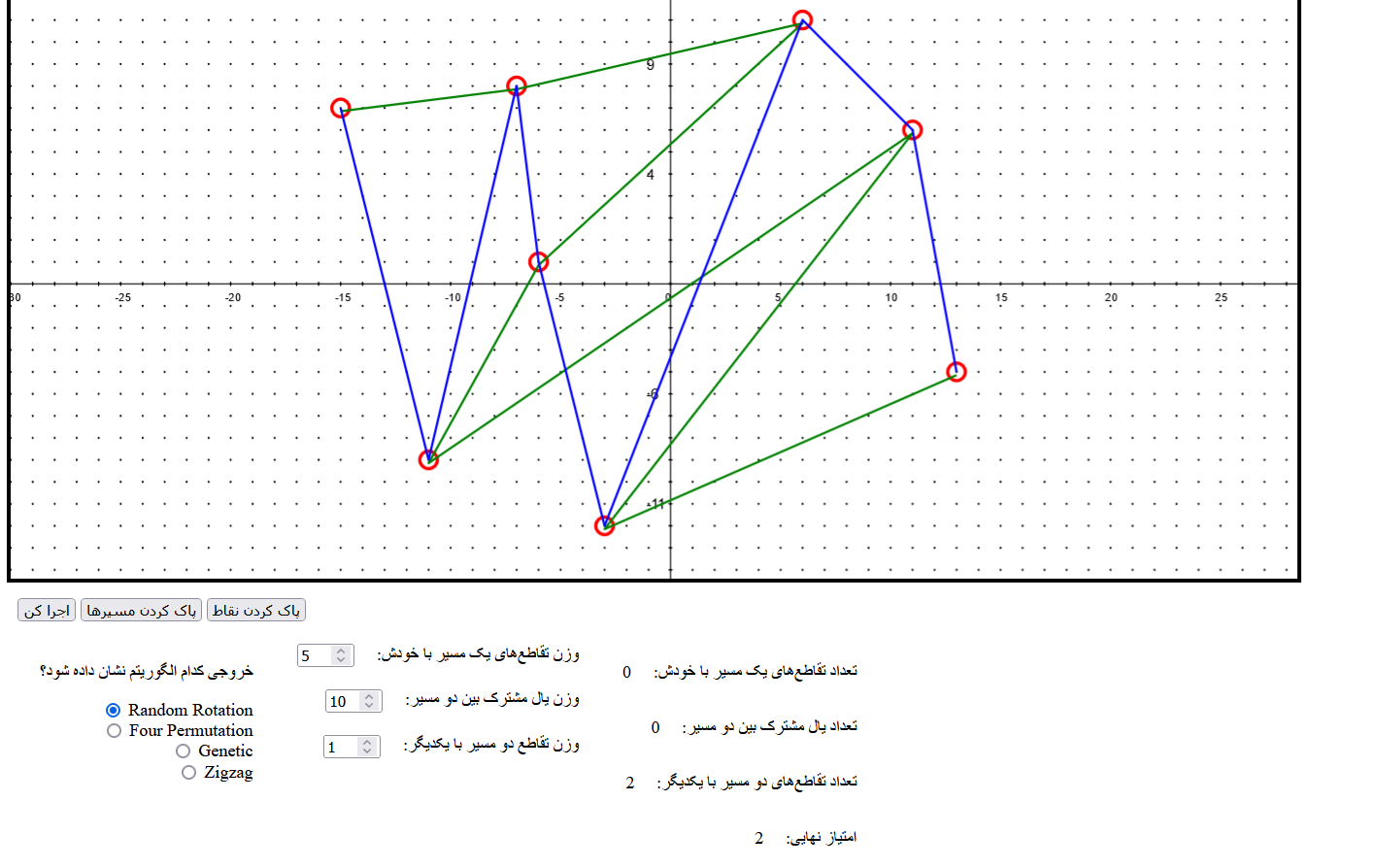
شکل 9: context دیاگرام سامانه نرم‌افزاری

۳.۲ نحوه استفاده

با توجه به شکل زیر، ابتدا کاربر با کلیک بر روی صفحه مختصات، نقاط ورودی را وارد می‌کند. سپس الگوریتم‌ مورد نظر و وزن دلخواه هر کدام از سه زیرمعیار خوانایی را مشخص می‌کند و دکمه «اجرا» را می‌زند (شکل فلان). همچنین کاربر می‌تواند با زدن دیگر دکمه‌ها، نقاط یا مسیر همیلتونی خروجی داده شده را پاک کند تا ورودی دیگری به سامانه بدهد.



با این کار، درخواستی ایجاد می‌شود و بعد از چند ثانیه، خروجی بدست آمده به همراه میزان خوانایی آن به کاربر نمایش داده می‌شود.



فصل چهارم

آزمایش‌ها و نتایج

در این فصل به بررسی نتایج آزمایش الگوریتم‌ها با هم می‌پردازیم. ابتدا شرایط و معیارهای سنجش را تعریف می‌کنیم. سپس با تعریف ساده‌تر خوانایی شروع می‌کنیم و الگوریتم‌ها را با هم مقایسه می‌کنیم. بعد از آن تعریف را کامل‌تر کرده و نتیجه‌گیری نهایی را انجام می‌دهیم.

۱.۴ شرایط و معیارهای سنجش

مقایسه الگوریتم‌ها از سه نظر کیفیت خروجی، میانگین سرعت اجرا و سادگی پیاده‌سازی صورت می‌گیرد. در قسمت خروجی، به هر کدام از سه زیرمعیار تقاطع یک مسیر با خود، یال مشترک بین دو مسیر و تقاطع دو مسیر با یکدیگر وزنی نسبت داده و میانگین جمع وزن‌دار این سه معیار بر روی تست‌های مختلف، میزان ناخوانایی را می‌سازد. باید دقت شود که در واقع، هر چه میزان ناخوانایی کمتر باشد، مسیرهای پیدا شده مسیرهای بهتری هستند.

برای انجام آزمایش‌ها از ۱۰۰ تست‌ که به صورت تصادفی تولید شده‌اند، کمک گرفته‌ایم و عملکرد الگوریتم‌ها را در آن‌ها میانگین گرفته‌ایم. تست‌ها در سه گروه گراف‌های ۴ تا ۱۵ راسی، ۱۶ تا ۳۰ راسی و ۳۱ تا ۶۰ راسی گروه‌بندی شده‌اند (با توجه به ابعاد جدول مختصات سامانه، بعید است کاربر سامانه نرم‌افزاری بیشتر از ۶۰ نقطه ورودی بدهد). لازم به ذکر است که الگوریتم «جایگشت چهارتایی» فقط برای بازه ۴ تا ۱۵ راسی کار می‌کند و در آزمایش دو گروه دیگر حضور ندارد. (تکمیل آینده: چرا ۱۰۰ کافیه)

نکته مهم دیگر این است که در این آزمایش‌ها بیشترین وزن منفی به روی هم افتادن یال‌های دو مسیر و کمترین وزن منفی به تقاطع دو مسیر با یکدیگر داده شده‌است. این وزن‌دهی بر اساس مشاهدات نویسنده و سنجش فردی خوانایی برای خود او می‌باشد. پس این وزن‌دهی بیشتر ذهنی (subjective) بوده و بسته به افراد مختلف امکان تفاوت‌هایی وجود دارد. (تکمیل آینده: نظر فردی نویسنده یه جوریه)

در آخر لازم به ذکر است که با توجه به اینکه این الگوریتم‌ها در سامانه نرم‌افزاری استفاده می‌شوند و کاربر با آن‌ها کار می‌کند، نمی‌توان فرصت زیادی به الگوریتم‌ها برای بهینه کردن جواب‌هایشان داد. از این رو هر الگوریتم به طور میانگین، تقریبا ۳۰ ثانیه فرصت خروجی دادن دارد.

۲.۴ خوانایی با در نظر گرفتن وزن برابر برای زیرمعیارها

در آزمایش‌های این قسمت به زیرمعیارهای تقاطع یک مسیر با خود، یال مشترک بین دو مسیر و تقاطع دو مسیر با یکدیگر وزن یکسان و برابر ۱ داده شده‌است.

۱.۲.۴ گراف‌های ۴ تا ۱۵ راسی

همانطور که در جدول مشاهده ‌می‌شود، در گراف‌های ۴ تا ۱۵ راسی، عملکرد الگوریتم ژنتیک از بقیه بهتر است. الگوریتم چرخش نقاط و جایگشت‌ چهارتایی بعد از ژنتیک به ترتیب الگوریتم‌های خوبی هستند اما الگوریتم زیگ‌زاگ، به خاطر تقاطع زیاد دو مسیر با هم عملکرد بسیار بدتری دارد.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| نام الگوریتم | میانگین زمان اجرا | میانگین تقاطع با خود | میانگین یال مشترک | میانگین تقاطع مسیرها | میانگین ناخوانایی |
| جایگشت چهارتایی | 1523ms | 1.11 | 0.83 | 0.99 | 2.93 |
| ژنتیک | 1432ms | 0.3 | 0.38 | 0.14 | 0.82 |
| چرخش نقاط | 369ms | 0 | 0.15 | 0.53 | 1.68 |
| زیگ‌زاگ | 1ms | 0.04 | 0 | 14.38 | 14.42 |

جدول 1: نتیجه تست بر روی گراف‌های ۴ تا ۱۵ راسی

۲.۲.۴ گراف‌های ۱۶ تا ۳۰

الگوریتم زیگ‌زاگ باز هم بسیار سریع‌تر از دو الگوریتم دیگر است اما خوانایی آن، نسبت به حالت قبلی بسیار کمتر از دو الگوریتم دیگر شد. همچنین اختلاف الگوریتم‌های ژنتیک و چرخش نقاط در این حالت، به نسبت حالت قبل، کمتر شده‌است ولی همچنان الگوریتم ژنتیک کمی بهتر است.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| نام الگوریتم | میانگین زمان اجرا | میانگین تقاطع با خود | میانگین یال مشترک | میانگین تقاطع مسیرها | میانگین ناخوانایی |
| ژنتیک | 5478ms | 3.39 | 1.35 | 1.62 | 6.36 |
| چرخش نقاط | 1726ms | 0 | 4.93 | 1.94 | 6.87 |
| زیگ‌زاگ | 1ms | 0.21 | 0 | 74.27 | 74.48 |

جدول 2: نتیجه آزمایش بر روی گراف‌های ۱۶ تا ۳۰ راسی

۳.۲.۴ گراف‌های ۳۱ تا ۶۰

همانطور که پیداست، الگوریتم زیگ‌زاگ از بقیه سریع‌تر است. همچنین الگوریتم چرخش نقاط در تعداد راس بالا بهتر از الگوریتم ژنتیک عمل کرده‌است.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| نام الگوریتم | میانگین زمان اجرا | میانگین تقاطع با خود | میانگین یال مشترک | میانگین تقاطع مسیرها | میانگین ناخوانایی |
| ژنتیک | 24395ms | 25.4 | 4.28 | 18.73 | 48.41 |
| چرخش نقاط | 6788ms | 0 | 13.59 | 5.03 | 18.62 |
| زیگ‌زاگ | 1ms | 0.38 | 0 | 301.73 | 302.11 |

جدول 3: نتیجه آزمایش بر روی گراف‌های ۳۱ تا ۶۰ راسی

۳.۴ تعریف کامل‌تر خوانایی

با توجه به اینکه در عمل، تاثیر زیرمعیارهای مختلف روی خوانایی یکسان نمی‌باشد، برای آزمایش‌های این قسمت، وزن زیرمعیارهای تقاطع یک مسیر با خودش، یال مشترک بین دو مسیر و تقاطع دو مسیر با یکدیگر را به ترتیب ۵، ۱۰ و ۱ داده‌ایم.

۱.۳.۴ گراف‌های ۴ تا ۱۵ راسی

طبق جدول، خوانایی الگوریتم ژنتیک و چرخش نقاط کمی از دو الگوریتم دیگر بهتر است. همچنین زمان اجرای الگوریتم‌ زیگ‌زاگ بسیار کمتر از بقیه است.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| نام الگوریتم | میانگین زمان اجرا | میانگین تقاطع با خود | میانگین یال مشترک | میانگین تقاطع مسیرها | میانگین ناخوانایی |
| جایگشت چهارتایی | 1252ms | 0.41 | 0.62 | 1.72 | 9.97 |
| ژنتیک | 1619ms | 0.12 | 0 | 0.81 | 1.41 |
| چرخش نقاط | 331ms | 0 | 0.05 | 2.85 | 3.35 |
| زیگ‌زاگ | 1ms | 0.02 | 0 | 12.44 | 12.54 |

جدول 4: نتیجه آزمایش بر روی گراف‌های ۴ تا ۱۵ راسی

۲.۳.۴ گراف‌های ۱۶ تا ۳۰ راسی

طبق جدول زیر، الگوریتم‌های ژنتیک و چرخش نقاط نسبتا به هم نزدیک‌تر هستند ولی الگوریتم زیگ‌زاگ خوانایی خیلی پایین‌تری دارد. از نظر سرعت اجرا نیز الگوریتم زیگ‌زاگ بسیار سریع‌تر از دو الگوریتم دیگر است.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| نام الگوریتم | میانگین زمان اجرا | میانگین تقاطع با خود | میانگین یال مشترک | میانگین تقاطع مسیرها | میانگین ناخوانایی |
| ژنتیک | 5813ms | 3.48 | 0.01 | 5.85 | 23.35 |
| چرخش نقاط | 1686ms | 0 | 0.27 | 16.63 | 19.33 |
| زیگ‌زاگ | 1ms | 0.04 | 0 | 70.5 | 70.07 |

جدول 5: نتیجه آزمایش بر روی گراف‌های ۱۶ تا ۳۰ راسی

۳.۳.۴ گراف‌های ۳۱ تا ۶۰ راسی

طبق معمول، الگوریتم زیگ‌زاگ بسیار سریع‌تر از بقیه الگوریتم‌هاست. نکته جالب این است که در تعداد نقاط بالا، الگوریتم ژنتیک خروجی بسیار بدتری از خود نشان ‌می‌دهد به طوری که فاصله بسیاری با الگوریتم چرخش نقاط می‌گیرد و حتی به الگوریتم زیگ‌زاگ نزدیک ‌می‌شود.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| نام الگوریتم | میانگین زمان اجرا | میانگین تقاطع با خود | میانگین یال مشترک | میانگین تقاطع مسیرها | میانگین ناخوانایی |
| ژنتیک | 24300ms | 27.71 | 0.31 | 35.71 | 177.36 |
| چرخش نقاط | 6740ms | 0 | 2.46 | 42.32 | 66.92 |
| زیگ‌زاگ | 1ms | 0.38 | 0 | 301.73 | 303.63 |

جدول 6: نتیجه آزمایش بر روی گراف‌های ۳۱ تا ۶۰ راسی

فصل پنجم

جمع‌بندی و کارهای آتی

در فصل‌های گذشته، ابتدا کمی درباره مسئله کشیدن خوانای گراف‌ها صحبت کردیم. سپس سامانه نرم‌افزاری مقایسه الگوریتم‌ها را معرفی کردیم. بعد راجع به الگوریتم‌های مختلف حل مسئله مذکور صحبت کردیم و آن‌ها را با هم مقایسه کردیم. در ادامه به کارهایی که می‌توان در آینده روی این مسئله انجام می‌پردازیم.

۱.۵ کارهای آینده

در قسمت‌های مختلفی از مسئله مورد نظر می‌توان کارهای مختلفی انجام داد که اینجا به طور مختصر اشاره می‌شود.

## فهرست گزارش، فهرست شكل‌ها و فهرست جداول

اگر از اين الگو براي تهيه پایان نامه استفاده كنيد، پس از اتمام يا در حين تكميل آن مي‌توانيد با راست‌كليك روي فهرست فعلي، آن را به‌روز كنيد (توسط گزينه update field). فهرست جداول و اشكال نيز به‌همين صورت قابل به‌روز‌شدن مي‌باشد.

## سربرگ و ته‌برگ (Header and Footer)

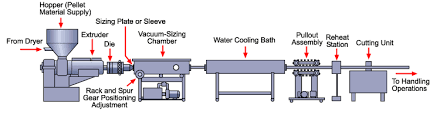
سربرگ و ته‌برگ را مي‌توان از منوي Insert انتخاب كرد‌.‌ به قاب‌هايي كه در قسمت بالا و پايين باز مي‌شود، به‌ترتيب سربرگ و ته‌برگ گفته مي‌شود‌.‌ در سمت راست سربرگ، عنوان هر فصل نوشته مي‌شود‌. در زير اين دو قسمت يك خط پررنگ قرار مي‌گيرد ‌و در ته‌برگ، شماره صفحه نوشته می شود.

## جداول، منحني‌ها، شكل‌ها

جداول و اشكال هر فصل بايد از شماره 1 به‌همراه شماره فصل مربوطه شماره‌گذاري‌ شود و در داخل متن (بلافاصله پس از بندي كه به آن شكل يا جدول ارجاع داده شده) آورده شود‌.‌ در متن بايد به تمامي جداول و شكل‌ها ارجاع داده شود‌.‌ در عنوان جداول و شكل‌ها نام كميت يا پديده مورد مطالعه ذكر مي‌شود.‌ عنوان جداول در بالای جدول و عنوان شکل در پایین ان ذکر گردد. (مشابه جدول 4-5 و شکل 4-1)

جدول4‌-‌5 عنوان جدول.

|  |  |  |
| --- | --- | --- |
| كار (WU) | زمان (ثانيه) | روش |
| 37/85 | 54 | چند شبكه‌اي |



شكل4‌-1 فرايند كواكستروژن.

## ارجاع به جداول، شكل‌ها، روابط، مراجع و بخش‌ها

* هرگز نبايد يك شكل يا جدول پيش از معرفی آن، در متن ظاهر شود‌.‌
* بين شماره شكل (يا جدول) و كلمه پیش از آن (شكل يا جدول) بايد حتماً يك فاصله قرار گيرد‌.‌ به‌عنوان مثال «شكل 2‌‌-‌‌2» صحيح، و «جدول2‌‌-‌‌2» ناصحیح است‌.‌
* براي ارجاع به شكل‌ها و جداول نبايد از پرانتز استفاده شود‌.‌ به‌عنوان مثال «جدول (2‌‌-‌‌2)» غلط است‌.
* ارجاع به روابط رياضی همواره بايد پس از درج خود آن روابط در متن صورت گيرد‌.
* برای ارجاع به روابط رياضی از شماره آنها در داخل پرانتز استفاده مي‌شود؛

# 

# فصل پنجم بررسي ساختار پایان نامهبررسي ساختار پایان نامه

به‌منظور استفاده از شيوه‌اي مناسب در تهيه يك پایان نامه، بايد روندي خاص دنبال شود. اين روند ممكن است تا حدودي شبيه به يك چک لیست ‌باشد، اما تبعيت از اين روند سبب مي‌شود در پايان گزارش‌نويسي، نيازي به صرف زمان فراوان براي ويرايش نهايي پایان نامه نباشد. از اين رو و به دليل اينكه ويرايش گزارش پایان نامه نياز به حوصله كافي دارد و در صورت استمرار، باعث كاهش حساسيت در رعايت ملزومات مي‌شود، به همین دلیل انجام ويرايش در پايان هر فصل، توصيه مي‌شود.

## تنظیم بندها

بند اول كه در زير عنوان آورده مي‌شود، از اول خط شروع شده و ساير بندها با 5 فاصله خالي شروع   
مي شوند، تصمیم‌گیری در مورد بندها و تنظیم آنها، يكي از ملزومات اصلي يك گزارش پایان نامه است. به‌منظور انجام هوشمندانه اين كار، از تعريف خودكار فاصله‌دهي (Indentation) براي شروع بند جديد در الگو خودداري شده‌است. علاوه بر اين، با توجه به اينكه در گزارش‌نويسي فني، روابط رياضي زيادي مورد استفاده قرار مي‌گيرد، خودكارشدن فاصله‌دهي، باعث مي‌شود كه پس از هر رابطه، بند جديد در‌نظر گرفته شود كه وضعيت مطلوبي نمي‌باشد. بنابراين حين نوشتن، لازم است كه نويسنده به‌صورت هوشمندانه با قراردادن 5 فاصله خالي در ابتداي هر بند، آن را متمايز كند (به بند بعد توجه كنيد!).

بايد توجه داشت كه از قرار‌دادن شكل، در وسط يك بند خودداري ‌شود. بنابراين شكل‌ها بايد در انتهاي بند مربوطه آورده شود و حتما پس از شكل بايد بند جديد و با فاصله‌دهي بيان‌شده براي بند، شروع شود. در مجموع، به‌جز بندي كه بلافاصله پس از سرعنوان مي‌آيد، همواره يك بند بايد با 5 فاصله خالي، آغاز شود تا قابل تشخيص باشد.

به‌منظور زيبايي بيشتر پایان نامه، توصيه مي‌گردد در بين يك بند، شكل و يا جدول قرار نگيرد. با اين وجود پس از قرار‌گرفتن شكل يا جدول در ميان بندهاي يك سر‌فصل، بند پس از آن بدون فاصله ابتدايي آغاز مي‌گردد. چنانچه فرمولي نيز در ميان يك بند قرار مي‌گيرد، پس از درج فرمول، بند از ابتداي خط آغاز مي‌شود و نيازي به 5 فاصله خالي نيست.

### بررسي كيفيت شكل و تطابق عنوان آن

يكي از موارد مهم بررسي عناوين و تطابق آنها با شكل موردنظر مي باشد. همزمان با بررسي عنوان شكل، مي‌توان كيفيت آن را نيز مدنظر قرار داد.

### بررسي كيفيت جدول و تطابق عنوان آن

در اين بررسي لازم است كه بين محتويات جدول و عنوان آن تطابق برقرار باشد. هم‌زمان با بررسي عنوان جدول، مي‌توان كيفيت آن را نيز مدنظر قرار داد.

## به‌روز‌رساني مراجع

معمولا هنگام نوشتن پایان نامه به‌دليل توجه به متن مورد نگارش، ارجاع‌دهي مناسب مورد غفلت قرار مي‌گيرد. پس از اتمام نگارش مي‌توان براي هر فصل به بررسي ارجاع‌دهي پرداخت. در اين فعاليت لازم است تا با دقت فراوان بندهاي مختلف را مورد بررسي قرار داد و درصورت استفاده از مرجع خاصي، به آن ارجاع داده شود. در گزارش‌هاي فني لازم است كه روابط مهم و همچنين روابطي كه استخراج آن به‌صورت كامل در پایان نامه ارائه نشده‌است، ارجاع‌دهي شود. بهتر است در ارجاع‌دهي روابط، شماره رابطه مورد استفاده در مرجع موردنظر نيز ذكر گردد. مثلا (رابطه (3-2)، [3]). در بخش مراجع لازم است كه مراجع به‌ترتيب حضور در متن آورده شوند. مي‌توان اين مساله را در هر فصل به‌روز‌رساني كرد. اگر چه به‌روز‌رساني در انتهاي ويرايش تمامي فصول كفايت مي‌كند.

## صفحه‌بندي

پس از پايان اقدامات ويرايشي مختلف، مي‌توان با مرور كلي پروژه پایانی، نسبت به اصلاح صفحه‌بندي اقدام كرد. مثلا گاهي با جابه‌جاكردن يك شكل يا كوچك و بزرگ‌نمودن آن، مي‌توان صفحه‌بندي بهتري ارايه كرد. توجه داشته باشيد، با توجه به اينكه پس از صفحه‌بندي احتمالا شماره صفحات تغيير مي‌كند، اين اقدام پیش از بررسي سربرگ‌ها كه در بخش بعد تشريح مي‌شود، انجام شود.

## سربرگ و ته‌برگ‌ها

با توجه به استفاده از ابزار «Insert › Break» ، در پايان هر فصل، لازم است كه در آغاز هر فصل، موارد زير چك شود.

* عنوان فصل

عنوان فصل، در قسمت سمت راست بالاي صفحه قرار دارد و لازم است كه در هر فصل اصلاح شود.

* شماره صفحه

شماره صفحه، در پائين و وسط هر صفحه قرار دارد و لازم است در هر صفحه چک شود.

# فصل ششم جمع‌بندي و نتيجه‌گيري و پیشنهاداتجمع‌بندي و نتيجه‌گيري

در پايان گزارش‌هاي علمي و فني لازم است كه جمع‌بندي يا نتيجه‌گيري نهايي ارائه شود. در اين موارد مي‌توان آخرين فصل پایان نامه كه پیش از مراجع قرار مي‌گيرد را به اين امر اختصاص داد.

**پیشنهادات**

در این بخش پیشنهاداتی که محقق جهت ادامه تحقیقات دارد ارایه می‌گردد. دقت شود که پیشنهادات باید از تحقیق انجام شده و نتایج ان حاصل شده باشد و از ذکر جملات کلی باید پرهیز کرد.

تکمیل آینده برای یال‌های مشترک طول مسیر رو بذاریم تو ناخوانایی

**منابع و مراجع**

[**https://github.com/Arash-codedev/openGA**](https://github.com/Arash-codedev/openGA)

**http://paperjs.org/**

**https://www.researchgate.net/publication/320944800\_openGA\_a\_C\_Genetic\_Algorithm\_library**

|  |  |  |
| --- | --- | --- |
| [1] | نام خانوادگی، نام (مولفان و مترجمان)؛ *عنوان اصلی كتاب*؛ عنوان فرعی كتاب (جزئيات عنوان كتاب در صورت وجود داخل پرانتز)، نام ساير افراد دخيل در تاليف يا ترجمه، ناشر، محل انتشار، شماره جلد، شماره ويرايش، سال انتشار به عدد‌.‌ | |
| [2] | منهاج، محمد‌باقر؛ *هوش محاسباتی* (جلد اول: مبانی شبكه‌های عصبی)، انتشارات دانشگاه صنعتی امير‌كبير، تهران، ويرايش اول، 1379‌.‌ | |
| [3] | نام خانوادگی، نام؛ نام‌‌خانوادگی و نام مولف دوم؛ مولف سوم؛ "عنوان مقاله به‌صورت عادي و داخل گيومه"، *نام كامل مجله به صورت ايتاليك*، شماره دوره يا جلد، شماره مجله، شماره صفحات، سال انتشار‌.‌ | |
| [4] | نام خانوادگی، نام مجری؛ *عنوان طرح پژوهشی به‌صورت ايتاليك*، شماره ثبت، نام كامل محل انجام و سفارش دهنده، سال انجام طرح‌.‌ | |
| [5] | نام شركت/ نام فرد؛ عنوان صفحه؛ *آدرس اينترنتي*. | |
|  | | [6] Book authors’ names; *Book Title in Italic* (and the title components, if any), Edition number, Publisher, Date of publish. | |
|  | | [7] Van de Vegte, J.; *Feedback Control Systems*, 2nd Edition, Prentice Hall, 1990. | |
|  | | [8] Authors’ names separated by comma-dots; “The paper title in Regular Times New Roman 12pt”, *Paper Address in Italic*, Publishing Place, paper page, Year of Publish. | |
|  | | [9] Safonov, M.; “Stability margins of diagonally perturbed multivariable feedback systems”, *IEEE Proceedings*, Part D, p. p. 251-256, Nov. 1982. | |
|  | | [10] Company Name/ Person Name; Page Title; *Internet Address*. | |

**يا بصورت الفبايي**

Gibson R.F "Principles of Composite Material Mechanics" Holman

J.P, Lloyd J.R (editors) McGraw-Hill (1994).

http:// 213.176.8.27/DL/Asklibraryian/FAQ/Theses.html.

Kantz H, Schreiber T "Nonlinear Time Series Analysis" 2nd ed.

Cambridge, Cambridge University Press (2004).

Minkowycz W.J, Roblach N.J, France D.M " Dynamic of Inert Gas

Bubbles in Force Convective Systems " Int J of Heat and Mass Transfer

Vol.24 (1981) PP 1433-1441.

Pagano N.J "Analysis of the Flexural Test of Bidirectional

Composites" J Composite Matter 1 (1967) pp 336-342.

Smith J.E "Biotechnology" 4th Ed New York, Cambridge University

Press (2004).

Soo S.L "Boundary Layer Motion of a Gas-Solid Suspension"

Proceeding of the Symposium on Interaction between Fluids and Particles,

Institute of Chemical Engineers Vol.1 (1962) PP 50-63.

Timoshinko S.P "Theory of Elastic Stability" McGraw-Hill, New

York (1985).

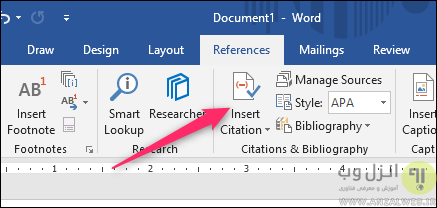
## یا مطابق دستور العمل زیر :

ایجاد منابع و اضافه کردن رونگاشت یا Citation به متن در ورد

۱- اشاره گر موس را بر روی بخشی از متن قرار دهید که می خواهید Citation یا استناد را به آن اضافه کنید.

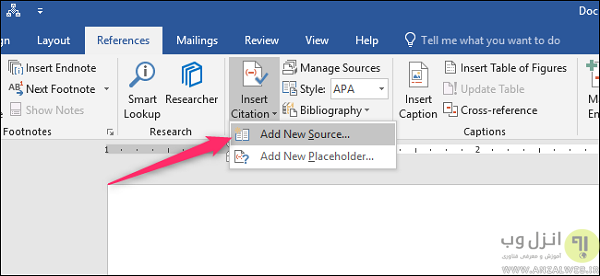
۲- به تب References در ریبون ورد بروید.

۳- روی Insert Citation کلیک کنید.



۴- منویی شامل همه منابعی که پیش از این اضافه کرده اید باز می شود.

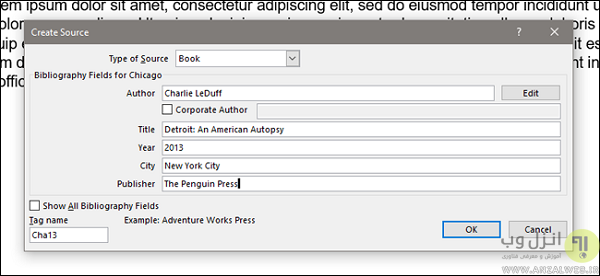
۵- برای اضافه کردن منبع جدید باید روی فرمان Add New Source کلیک کنید.



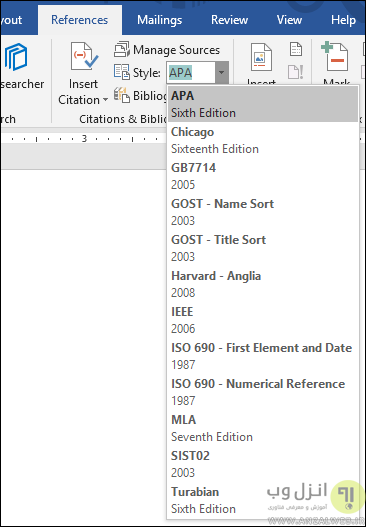
۶- در پنجره Create Source می توانید اطلاعات مربوط به هر منبع را وارد کنید.

۷- مقدار پیش فرض لیست Type of Source، کتاب(Book) است. به این معنی که این منبع در واقع یک کتاب می باشد. با باز کردن آن می توانید گزینه های دیگری را برای نوع منبع مورد استفاده انتخاب کنید.

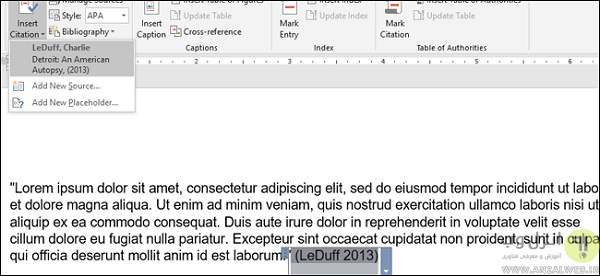
۸- می توانید برای این منبع یک نام برچسب (معمولا نسخه کوتاه تر نام منبع) یا دیگر اطلاعات را انتخاب کنید و در نهایت روی OK کلیک کنید.



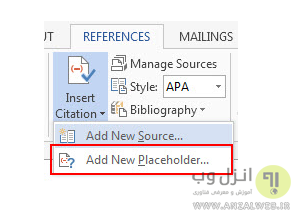
۹- به طور پیش فرض در نرم افزار ورد از روش نوشتن فهرست منابع به روش APA برای منابع استفاده می کتد. اگر به غیر از ایجاد **فهرست منابع به روش APA**شما از روش دیگری در سند خود استفاده می کنید می توانید روی گزینه Show All Bibliography fields کلیک کنید تا بتوانید از میان مواردی که به نمایش در می آید یکی از استایل ها را انتخاب کنید.



۱۰- با استفاده از مراحل بالا موفق به **درج منابع در ورد** شده اید. دفعه بعد که می خواهید از همین منابع در متن خود استفاده کنید کافی است تنها روی دکمه Insert Citation کلیک کنید. سپس منبع مورد نظر را از لیست انتخاب کنید تا به متن مورد نظر شما اضافه شود.



۱۱- ضمنا گاهی اوقات شما منبع اصلی و قابل استناد را دارید ولی تمامی اطلاعات مربوط به این منبع را در حال حاضر در اختیار ندارید. بنابراین شاید دلتان بخواهد فضایی خالی مربوط به این اطلاعات را در اختیار داشته باشد تا پس ار تکمیل شدن اطلاعات آن ها را وارد کنید. برای این کار می توانید ابتدا بر روی Insert Citation کلیک کرده و سپس گزینه Add New Placeholder را انتخاب کنید.

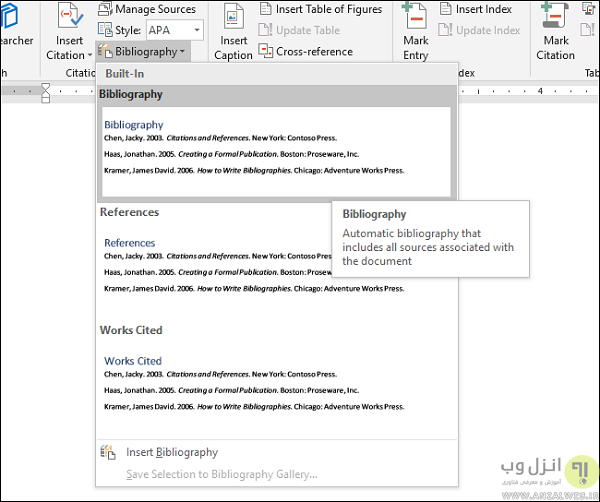


**ایجاد Bibliography یا فهرست منابع و ماخذ در مقاله ورد**

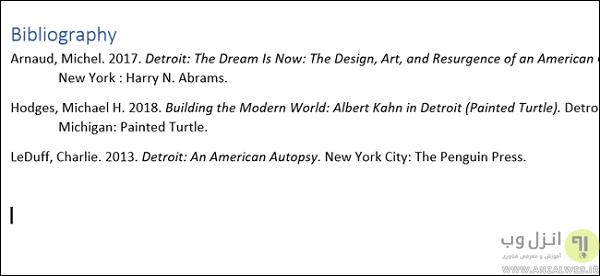
Bibliography به معنی ایجاد لیستی از منابع است. این لیست معمولا در انتهای سند قرار داده می شود. برای ایجاد این لیست طبق مراحل زیر عمل کنید:

۱- با استفاده از مسیر Layout > Breaks > Page Break یک صفحه جدید باز کنید.

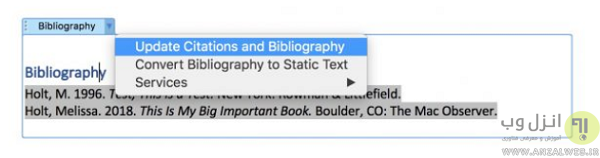
۲- به تب References رفته و روی دکمه Bibliography کلیک کنید.



۳- می توانید از بین استایل های پیش فرض دارای هدر، یکی از آن ها را انتخاب کنید. اما اگر استایلی بدون هدر می خواهید می توانید روی گزینه Insert Bibliography کلیک کنید. در این هنگام ورد همه منابعی را که از آن ها استفاده نموده اید طبق فرمت انتخابی شما به سند اضافه می کند.



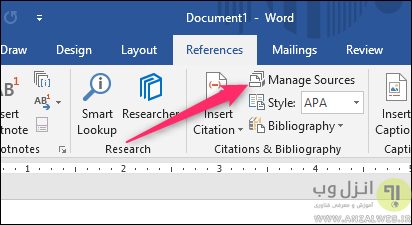
۴- اگر قبلا فهرستی از منابع را تهیه کردید و حالا می خواهید منابع دیگری را نیز به آن اضافه کنید می توانید روی آن کلیک کنید تا یک هدر پدیدار شود. با کلیک بر روی فلش گینه های مختلفی را خواهید دید. در اینجا گزینه Update Citations and Bibliography را انتخاب کنید تا فهرست منابع به روز رسانی شود.



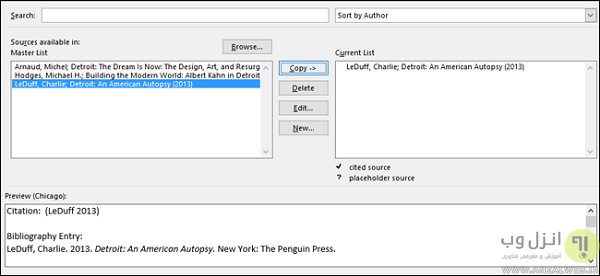
**پشتیبان گیری و بازیابی رفرنس ها در نرم افزار ورد**

ممکن است در حال نوشتن مقالات گوناگونی در یک زمینه خاص باشید و به همین دلیل نمی خواهید درگیر وارد کردن منابع تکراری در هر سند به صورت جداگانه باشید. خوشبختانه در ورد برای این موارد هم راه چاره ای پیش بینی شده است. هر زمان که منبع جدیدی را وارد می کنید ورد ان را در لیستی به نام Master source list ذخیره می کند. برای هر سند جدید شما می توانید به بازیابی منابع قدیمی از طریق این لیست بپردازید و ان ها را در سند جدید خود وارد کنید. برای این کار طبق مراحل زیر پیش بروید:

۱- در تب References روی دکمه Manage Sources کلیک کنید.



۲- پنجره ای شامل همه منابعی که تا حالا استفاده نموده اید باز می شود. روی یکی از منابع در سمت چپ پنجره کلیک کرده و سپس دکمه Copy را کلیک کنید. این کار را برای همه منابعی که نیاز دارید تکرار کرده و در نهایت OK را انتخاب کنید.

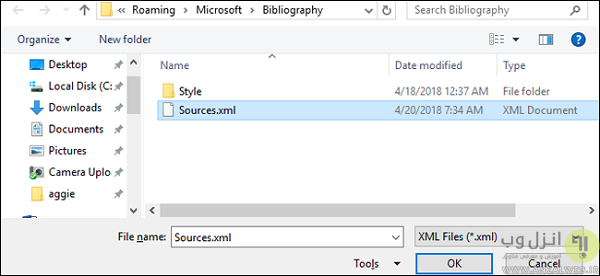


۳- اگر از تعداد زیادی منبع استفاده می کنید می توانید از طریق ابزاری در بالای این پنجره به جستجو میان آن ها بپردازید. این کار می تواند بر اساس عنوان، سال انتشار یا موارد دیگر انجام شود تا پیدا کردن منبع مورد نظر ساده تر شود.

اگر نیاز دارید که لیست منابع خود را به کامپیوتر دیگری انتقال دهید می توانید آن را به شکل یک فایل XML در مسیر زیر پیدا کنید:

C:\Users\username\AppData\Roaming\Microsoft\Bibliography

با کپی این فایل در کامپیوتر دیگری برنامه ورد را باز کرده و دکمه Manage Sources را انتخاب کنید تا با جستجوی این فایل آن را وارد ورد کنید.



ضمنا با گزینه هایی که با کلیک بر روی دکمه Manage Sources در اختیار شما قرار می گیرد می توانید به **درج و حذف رفرنس در ورد** یا ویرایش آن ها بپردازید.

# پيوست‌ها

موضوعات مرتبط با متن گزارش پایان نامه كه در يكی از گروه‌های زير قرار می‌گيرد، در بخش پيوست‌ها آورده شوند:

1- اثبات های رياضی يا عمليات رياضی طولانی‌.‌

2- داده و اطلاعات نمونه (های) مورد مطالعه (Case Study) چنانچه طولانی باشد‌.‌

3- نتايج كارهای ديگران چنانچه نياز به تفصيل باشد‌.‌

4- مجموعه تعاريف متغيرها و پارامترها، چنانچه طولانی بوده و در متن به انجام نرسيده باشد‌.‌

براي شماره‌گذاري روابط، جداول و اشكال موجود در پيوست‌ از ساختار متفاوتي نسبت به متن اصلي استفاده مي‌شود كه در زير به‌عنوان نمونه نمايش داده شده‌است.

(پ-1) 

جدول پ-1: شرح كد منبع بدنه اصلي يك كد رايانه‌اي.

|  |
| --- |
| 01 program AeroPack;  02 uses  03 Forms,  04 Unit1 in 'Unit1.pas' {Form1},  05 Dialogs,  06 Sysutils;  07 {$R \*.res}  08 begin  09 Application.Initialize;  10 Application.Title := 'AeroPack';  11 Application.CreateForm(TForm1, Form1);  12 if pos('/h',Form1.Switches)<>0 then  13 begin  14 Application.ShowMainForm:=False;  15 Form1.Visible:=False;  16 end;  در صورتيكه سوئيچ /h در رشته سوئيچ موجود باشد، متغير ShowMainForm و خصوصيت Visible فرم اصلي را برابر با False قرار مي‌دهد. نتيجه اين كار عدم نمايش فرم اصلي خواهد بود.  17 Application.Run;  18 end. |

# Abstract

This page is accurate translation from Persian abstract into English.

**Key Words:** Write a *3* to *5* KeyWords is essential.



Amirkabir University of Technology  
(Tehran Polytechnic)

… Department …

MSc or PhD Thesis

Title of Thesis

By

Name

Supervisor

Dr.

Advisor

Dr.

Month & Year