



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

رشته: علوم کامپیوتر

هوش مصنوعی و کارگاه - الگوریتم ژنتیک

نگارش

امیرمحمد کاوش

استاد راهنما

دکتر مهدی قطعی

استاد کارگاه

دکتر بهنام یوسفی مهر

خرداد/۱۴۰۴

## چکیده

مسئله فروشنده دوره گرد (TSP) یکی از مسائل کلاسیک در بهینه‌سازی است که هدف آن یافتن کوتاه‌ترین مسیر بین مجموعه‌ای از شهرهاست، به‌طوری‌که هر شهر فقط یک‌بار بازدید شود و مسیر به نقطه آغاز بازگردد. به‌دلیل پیچیدگی محاسباتی بالا، از الگوریتم‌های فراابتکاری مانند الگوریتم ژنتیک برای حل آن استفاده می‌شود. در این پروژه، الگوریتم ژنتیک برای حل TSP روی مجموعه‌ای شامل مختصات جغرافیایی ۳۸ شهر از کشور جیبوتی پیاده‌سازی شده است.

در این الگوریتم، با طراحی مناسب کروموزوم‌ها و تنظیم پارامترهایی مانند نرخ جهش، باز ترکیب، اندازه جمعیت و تعداد نسل، سعی شده است بین تنوع ژنتیکی و سرعت همگرایی تعادل برقرار شود. همچنین، سه استراتژی انتخاب (تورنمنت، رولت، رتبه‌ای)، باز ترکیب (تک‌نقطه، دونقطه، یکنواخت) و جهش (تعویض، معکوس‌سازی، آشفستگی) بررسی و مقایسه شده‌اند تا بهترین عملکرد الگوریتم حاصل شود.

واژه‌های کلیدی: الگوریتم ژنتیک، مسئله فروشنده دوره گرد، بهینه‌سازی

	چکیده.....	أ
	فصل اول مقدمه مقدمه.....	1
	فصل دوم الگوریتم ژنتیک.....	3
	الگوریتم ژنتیک.....	4
	2-1- کروموزوم ها در TSP.....	4
	2-2- ژن در TSP.....	4
	فصل سوم پارامترهای الگوریتم ژنتیک پارامترهای الگوریتم ژنتیک.....	5
	3-1- اندازه جمعیت.....	6
	3-2- تعداد نسل (تعداد اجرا).....	7
	3-4- نرخ جهش.....	9
	3-5- نرخ باز ترکیب.....	10
	3-6- استراتژی ها.....	11
	3-6-1- استراتژی انتخاب.....	11
11	3-6-1-1- تورنمنت.....	11
12	3-6-2-1- رولت.....	12
13	3-6-2-3- رنک.....	13
	3-6-2- مقایسه استراتژی های انتخاب.....	13
	3-6-2- استراتژی باز ترکیب.....	14
	3-6-2-1-.....	14
	3-6-2-2-.....	14
	3-6-2-3-.....	14
14	3-6-2-4- تک نقطه.....	14
15	3-6-2-5- دو نقطه.....	15
15	3-6-2-6- یکپارچه.....	15
	3-6-2-4- مقایسه استراتژی های باز ترکیب.....	15
	Abstract.....	19

## فصل اول

### مقدمه

## مقدمه

مسئله فروشنده دوره گرد (Problem Salesman Traveling) یا (TSP) یکی از معروف ترین مسائل بهینه سازی است. در این مسئله، هدف این است که یک فروشنده از هر شهر دقیقاً یک بار عبور کرده و در نهایت به شهر اولیه بازگردد، به گونه ای که هزینه یا مسافت کلی مسیر کمینه شود. این مسئله در حوزه های مختلفی مانند حمل و نقل، لجستیک، طراحی مدارهای الکترونیکی و مسائل بهینه سازی تولید کاربرد دارد.

TSP یک مسئله NP-hard است؛ به این معنا که برای تعداد زیادی از شهرها، یافتن جواب بهینه در زمان معقول غیرممکن است. با این حال، حل این مسئله به دلیل کاربردهای گسترده اش در دنیای واقعی اهمیت زیادی دارد. الگوریتم های ژنتیک به عنوان یک روش بهینه سازی برای حل مسائل NP-hard مانند TSP استفاده می شوند. این الگوریتم ها می توانند نتایج قابل قبولی برای مسئله TSP ارائه دهند، هرچند تضمینی برای یافتن راه حل بهینه ندارند.

ما در این مقاله مدلی پیاده کردیم، که به حل این مسئله از دیتاستی شامل مختصات جغرافیایی ۳۸ شهر از کشور جیبوتی پرداخته است. این داده ها برای مسئله فروشنده دوره گرد (TSP) طراحی شده اند و از پایگاه داده های آژانس ملی نقشه برداری (NIMA) و سیستم اطلاعات نام های جغرافیایی (GNIS) استخراج شده اند. فرمت داده ها به صورت طول و عرض جغرافیایی برای هر شهر است و تابع هزینه بصورت فاصله اقلیدسی تعریف شده.

## فصل دوم

## الگوریتم ژنتیک

## الگوریتم ژنتیک

الگوریتم ژنتیک یکی از روش‌های هوش مصنوعی است که بر پایه‌ی فرایند تکامل طبیعی بنا شده و برای حل مسائل بهینه‌سازی به کار می‌رود. در این الگوریتم، مجموعه‌ای از جواب‌های اولیه تولید می‌شود که به آن‌ها *population* (جمعیت) گفته می‌شود. هر جواب، مانند یک *chromosome* (کروموزوم)، از تعدادی *gene* (ژن) تشکیل شده که در مسئله‌هایی مثل TSP نشان‌دهنده ترتیب بازدید از شهرهاست. الگوریتم با اعمال عملگرهای انتخاب، *crossover* (ترکیب) و *mutation* (جهش) روی این جمعیت، نسل‌های جدیدی ایجاد می‌کند و به تدریج به سمت جواب‌های بهتر حرکت می‌کند. در مسئله فروشنده دوره‌گرد، الگوریتم ژنتیک تلاش می‌کند مسیری را پیدا کند که کمترین هزینه یا مسافت را داشته باشد، حتی اگر جواب صددرصد بهینه نباشد.

### 2-1- کروموزوم‌ها در TSP

هر جواب این مسئله به صورت دنباله‌ای مرتب از شهرهاست. ما هر شهر را به یک ایندکس نظیر می‌کنیم و در نهایت آن دنباله‌ای از شهرها که جواب یک مسئله است را گزارش می‌کنیم

### 2-2- ژن در TSP

با توجه به کروموزوم توصیف شده و دیتاست مورد بررسی ما در این پروژه، ما با کروموزوم‌هایی مواجه هستیم که از 38 ژن تشکیل شده‌اند. هر ژن یک عدد صحیح است که مقدار آن، شماره شهر مقصد است.

## فصل سوم

### پارامترهای الگوریتم ژنتیک

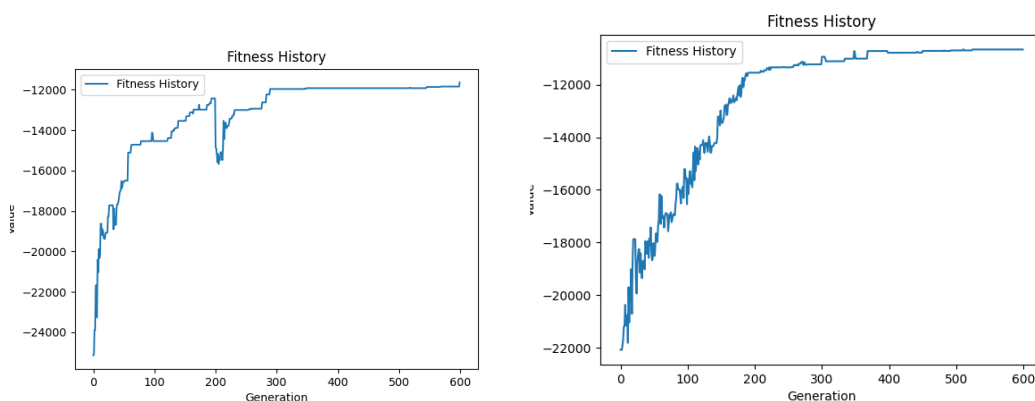


## پارامترهای الگوریتم ژنتیک

در الگوریتم ژنتیک، پارامترهایی مانند نرخ جهش، نرخ ترکیب، اندازه جمعیت و ... تأثیر مستقیمی بر عملکرد الگوریتم دارند. اگر این پارامترها نادرست تنظیم شوند، الگوریتم یا شبیه جستجوی تصادفی عمل می‌کند (در اثر جهش زیاد) یا در بهینه محلی گیر می‌افتد (در اثر انتخاب‌های تهاجمی و تنوع کم). تعادل میان اکتشاف و بهره‌برداری با استفاده از استراتژی‌های مناسب، ترکیب یکنواخت و جهش کنترل‌شده باعث پایداری و کارایی الگوریتم می‌شود.

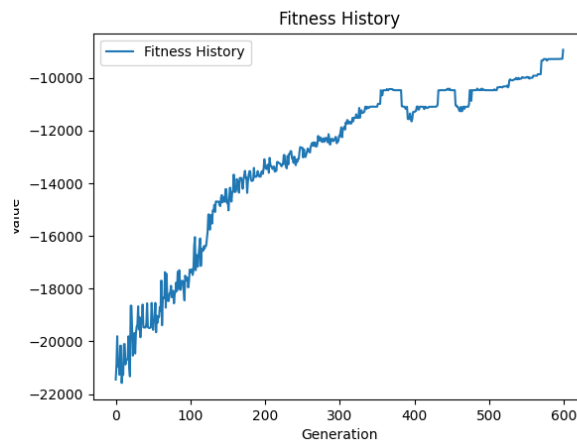
### 3-1- اندازه جمعیت

اندازه جمعیت اگر کم باشد، تنوع ژنتیکی پایین می‌آید و احتمال گیر افتادن در بهینه محلی زیاد می‌شود. اگر خیلی زیاد باشد، تنوع بیشتر می‌شود ولی زمان اجرا بالا می‌رود و ممکن است الگوریتم کند پیش برود و نویزهای زیادی ما را فریب دهند. اندازه مناسب، تعادلی بین کیفیت جواب و سرعت همگرایی ایجاد می‌کند. معمولاً اندازه جمعیت بین ۵ تا ۱۰ برابر تعداد ژن‌ها (در اینجا، شهرها) انتخاب می‌شود. برای ۳۸ شهر، جمعیت بین ۱۹۰ تا ۳۸۰ مناسب در نظر گرفته می‌شود. عدد ۲۵۰ در همین بازه و قابل قبول است.



تصویر 1 جمعیت 20 تایی

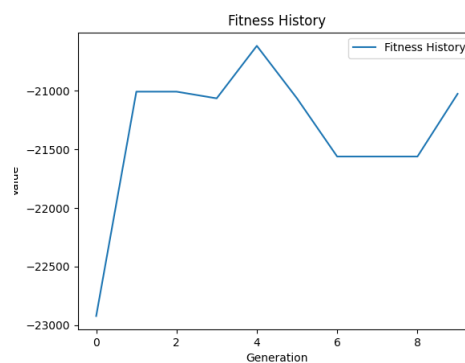
تصویر 2 جمعیت 250 تایی



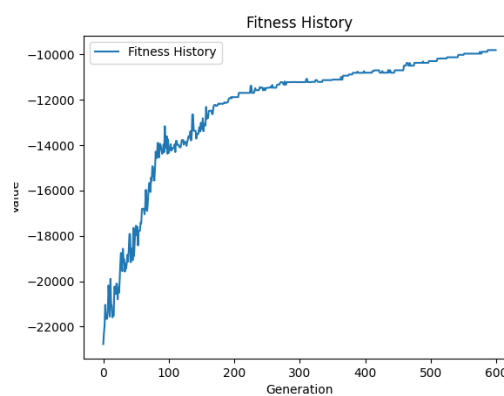
تصویر 3 جمعیت 700 تایی

## 2-3- تعداد نسل (تعداد اجرا)

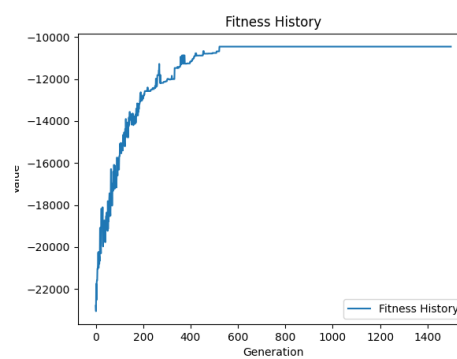
تعداد نسل کم باعث می‌شود الگوریتم فرصت کافی برای بهبود پاسخ نداشته باشد و در مراحل اولیه متوقف شود، که معمولاً منجر به جواب‌های ضعیف یا ناقص می‌شود. در مقابل، تعداد نسل زیاد امکان بهبود تدریجی و یافتن پاسخ‌های دقیق‌تر را فراهم می‌کند، اما اگر تنوع جمعیت حفظ نشود، ممکن است باعث همگرایی به پاسخ‌های تکراری یا بهینه محلی شود که از لحاظ زمانی هم بهینه نیست. در نتیجه، باید تعادلی میان تعداد نسل و حفظ تنوع ژنتیکی برقرار گردد تا الگوریتم از ظرفیت کامل بهره‌مند شود. برای چنین مسائلی معمولاً تعداد نسل‌ها بین ۵۰۰ تا ۱۰۰۰ انتخاب می‌شود تا الگوریتم فرصت کافی برای جستجو و بهبود پاسخ‌ها داشته باشد. عدد ۶۰۰ به اندازه‌ای زیاد هست که بتواند از طریق جهش و ترکیب‌های متنوع، جمعیت را به سمت جواب‌های بهتر هدایت کند، و در عین حال از نظر زمانی نیز مقرون به صرفه است.



تصویر 4 تعداد نسل 10



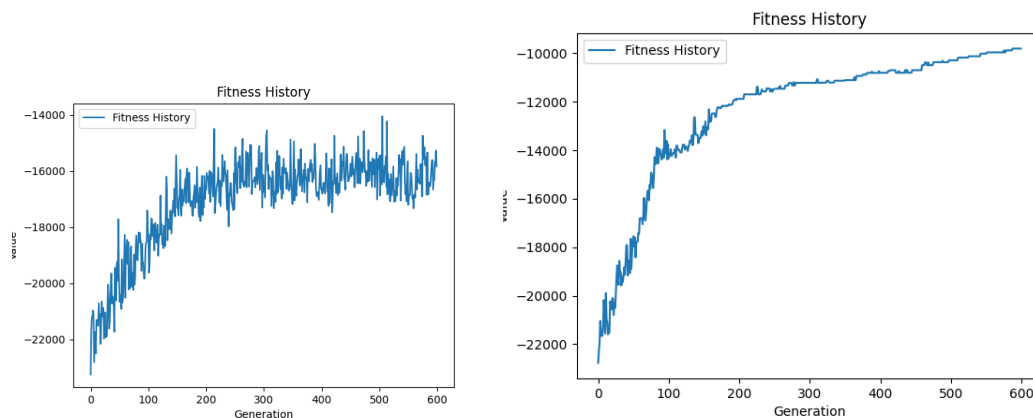
تصویر 5 تعداد نسل 600



تصویر 6 تعداد نسل 1500

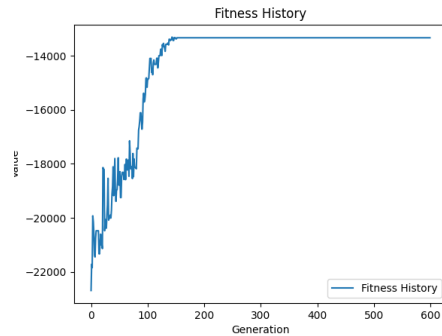
## 2-4- نرخ جهش

نرخ جهش (mutation rate) یکی از حساس‌ترین پارامترهای الگوریتم ژنتیک است که نقش مهمی در حفظ تنوع ژنتیکی جمعیت ایفا می‌کند. مقدار کم آن (مثلاً کمتر از ۰.۰۰۱) ممکن است باعث همگرایی زودهنگام و گیر افتادن در بهینه‌های محلی شود، چون تغییرات کافی برای کشف نواحی جدید صورت نمی‌گیرد. در مقابل، مقدار زیاد آن (بیش از ۰.۰۲) الگوریتم را به رفتار تصادفی شبیه می‌کند و روند جستجو را ناپایدار می‌سازد. مقدارهای میانه مانند ۰.۰۰۳ تا ۰.۰۰۱ معمولاً برای ایجاد تعادل بین اکتشاف و بهره‌برداری مناسب هستند.



تصویر 7 نرخ جهش 0.8

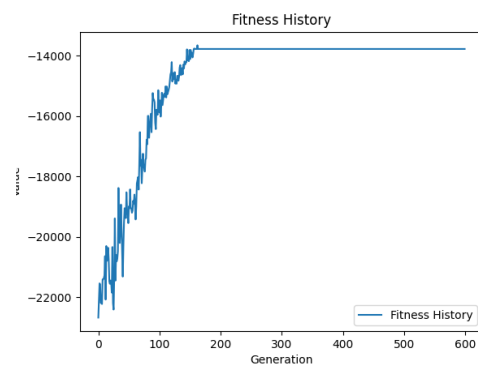
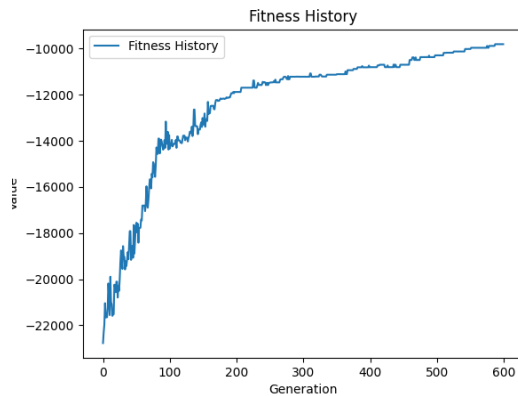
تصویر 8 نرخ 0.1



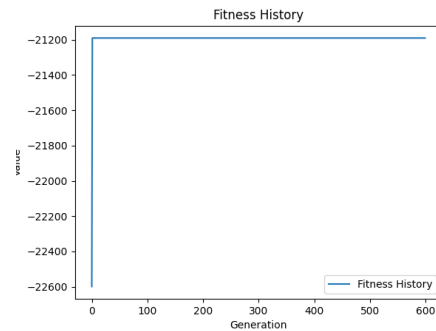
تصویر 9 نرخ جهش 0

## 2-5- نرخ باز ترکیب

نرخ باز ترکیب (crossover rate) تعیین می‌کند چند درصد از جمعیت با ترکیب والدین جدید تولید می‌شوند. مقدار پایین (مثلاً زیر ۰٫۶) ممکن است روند یادگیری را کند کند، چون ترکیب مؤثر ژن‌ها کمتر اتفاق می‌افتد. مقدار خیلی بالا (مثلاً بالای ۰٫۹) نیز ممکن است ساختارهای خوب موجود را بیش‌ازحد بشکند و باعث ناپایداری شود. بازه رایج و مؤثر معمولاً بین ۰٫۷ تا ۰٫۹ است. مقدار ۰٫۸ که رایج‌ترین انتخاب است، تعادلی مناسب بین حفظ ساختارهای خوب و تولید تنوع ایجاد می‌کند.



تصویر 10 نرخ 0.9 و 1



تصویر 11 نرخ 0.1

## 2-6- استراتژی ها

### 2-6-1- استراتژی انتخاب

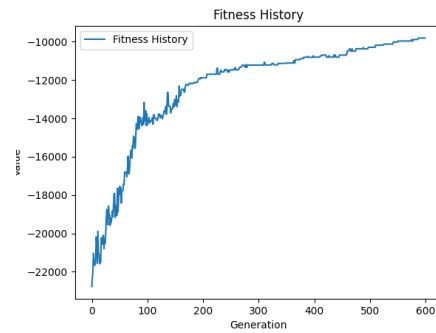
انتخاب در الگوریتم ژنتیک مرحله‌ای است که طی آن، افراد مناسب‌تر از جمعیت فعلی برای تولید نسل بعد انتخاب می‌شوند. هدف آن حفظ افراد خوب و افزایش احتمال تولید پاسخ‌های بهتر در نسل‌های آینده است. ما در این پروژه سه استراتژی را برای انتخاب والد در نظر گرفته ایم.

#### 2-6-1-1- تورنمنت

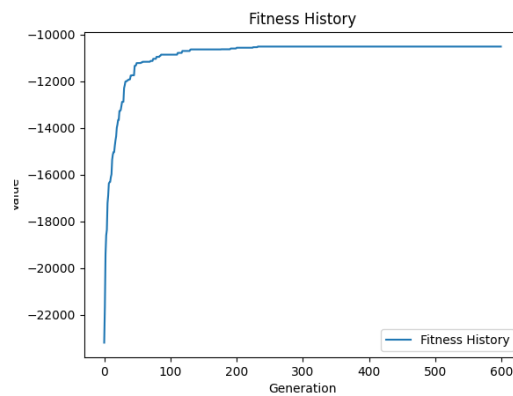
به اندازه سائز آن (مثلاً 3) کروموزوم از جمعیت‌مان به تصادف انتخاب می‌کنیم و بهترین آنها را بر اساس بالاترین مقدار تابع برازش، اعلام می‌کنیم.

#### سائز تورنمنت

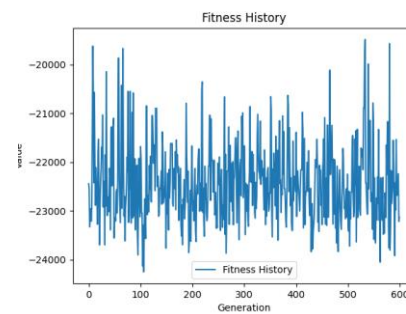
سائز تورنمنت معمولاً بین 2 تا 5 انتخاب می‌شود. مقدار پایین (مثلاً 2) باعث حفظ تنوع و کاهش فشار انتخاب می‌شود، در حالی که مقدار بالاتر (مثلاً 4 یا 5) فشار انتخاب را بیشتر کرده و الگوریتم سریع‌تر همگرا می‌شود، اما ممکن است تنوع جمعیت را کاهش دهد. مقدار 3 معمولاً تعادلی خوب بین این دو حالت فراهم می‌کند.



تصویر 12 سایز تورنمنت 3



تصویر 13 سایز تورنمنت 100 تنوع کم شد



تصویر 14 سایز 1 تنوع بیش از حد شد

## 2-1-6-2 رولت

در روش انتخاب رولت (Roulette Wheel Selection)، احتمال انتخاب هر فرد متناسب با مقدار برازندگی اوست؛ به گونه‌ای که افراد با برازندگی بیشتر شانس بالاتری برای انتخاب دارند. این روش بر

پایه‌ی انتخاب تصادفی وزن دار عمل می‌کند و شبیه چرخ رولت است که هر بخش آن متناسب با برازندگی افراد گسترده‌تر یا محدودتر است. اگرچه این روش می‌تواند افراد مناسب را به‌خوبی انتخاب کند، اما در حضور اختلاف زیاد در برازندگی یا نويز، پایداری کمتری دارد.

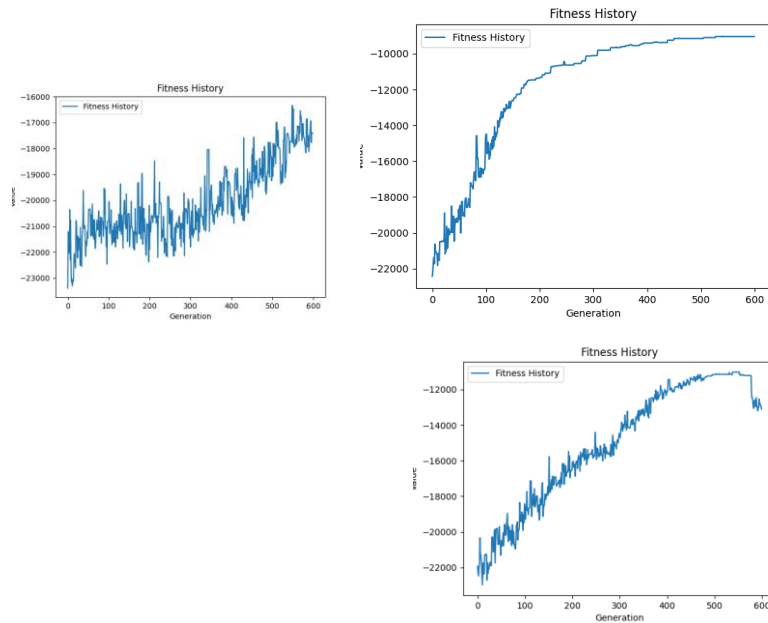
### 3-1-6-2- رنک

در روش انتخاب رتبه‌ای (Rank Selection)، افراد براساس رتبه‌بندی برازندگی‌شان مرتب می‌شوند، نه مقدار مطلق آن. سپس احتمال انتخاب هر فرد براساس رتبه‌اش تعیین می‌گردد، نه برازندگی واقعی. این کار باعث می‌شود فشار انتخاب یکنواخت‌تر شده و تأثیر افراد بسیار قوی یا بسیار ضعیف کاهش یابد. روش رتبه‌ای در جمعیت‌هایی با پراکندگی زیاد یا وجود مقادیر پرت، پایداری بیشتری ایجاد می‌کند.

### 2-6-2- مقایسه استراتژی‌های انتخاب

توضیح تکمیلی	سرعت همگرایی	حفظ تنوع ژنتیکی	استراتژی انتخاب
همگرایی آهسته ولی پایدار، مناسب برای مسائل پیچیده و چندپیکره	کم	زیاد	Rank Selection
تعادل بین تنوع و همگرایی، بستگی به اندازه تورنمنت دارد	متوسط (قابل تنظیم)	متوسط (قابل تنظیم)	Tournament
سریع ولی مستعد گیر افتادن در بهینه محلی، مناسب برای مسائل ساده زیاد	کم	کم	Roulette Wheel





تصویر 15 رولت و تورنمنت و رنک

## 2-6-3- استراتژی باز ترکیب

باز ترکیب (Crossover) یکی از مراحل کلیدی الگوریتم ژنتیک است که با ترکیب ژن‌های والدین، فرزندان جدیدی تولید می‌کند. هدف آن انتقال ویژگی‌های مفید از نسل قبل به نسل بعد و کاوش مؤثرتر فضای پاسخ با ایجاد تنوع ساختاری در جمعیت است.

## 2-6-3-4 تک نقطه

در باز ترکیب تک نقطه (One-point crossover)، یک نقطه تصادفی روی کروموزوم والدین انتخاب می‌شود، سپس بخش ابتدایی یکی از والدین با بخش انتهایی والد دیگر ترکیب می‌گردد تا دو فرزند جدید ایجاد شود. این روش ساده است و ساختار ژنتیکی والدین را تا حدی حفظ می‌کند.

### 2-6-3-5- دو نقطه

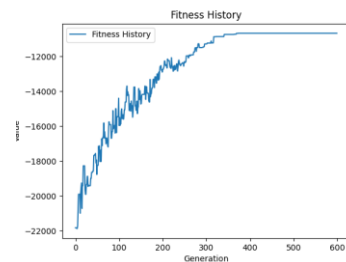
در باز ترکیب دو نقطه‌ای (Two-point crossover)، دو موقعیت تصادفی روی کروموزوم والدین انتخاب می‌شود و بخش میانی بین این دو نقطه از یکی از والدین با بخش‌های بیرونی از والد دیگر جابه‌جا می‌شود. این روش نسبت به تک‌نقطه‌ای تنوع بیشتری در فرزندان ایجاد می‌کند.

### 2-6-3-6- یکپارچه

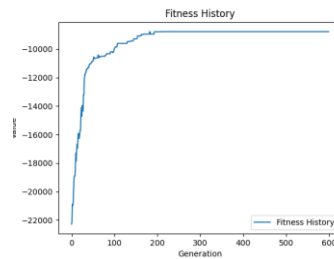
در باز ترکیب یکپارچه (Uniform crossover)، برای هر ژن به صورت مستقل با احتمال مشخصی (مثلاً ۰.۵) تصمیم گرفته می‌شود که از کدام والد به فرزند منتقل شود. این روش بیشترین تنوع را در ساختار فرزندان ایجاد می‌کند و ترکیب‌های ژنتیکی گوناگونی را ممکن می‌سازد.

### 2-6-4- مقایسه استراتژی‌های باز ترکیب

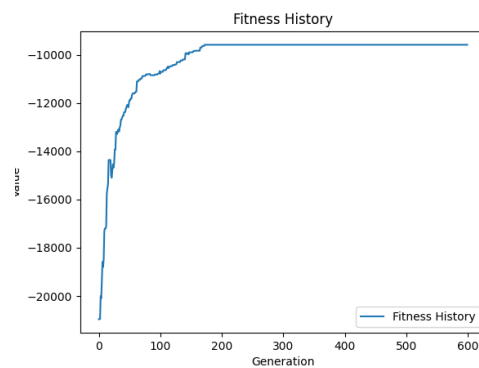
توضیح تکمیلی			
تولید شده	همگرایی	باز ترکیب	تنوع ژنتیکی — سرعت استراتژی
ساختار والد را بیشتر حفظ می‌کند، همگرایی سریع‌تر	کم	زیاد	One-point
تعادل بین حفظ ساختار و ایجاد تنوع	متوسط	متوسط	Two-point
بیشترین تنوع، ولی همگرایی آهسته‌تر و نیازمند نسل‌های بیشتر	زیاد	کم	Uniform



تصویر 16 یونیفرم



تصویر 17 تک نقطه



تصویر 18 دونقطه

## 5-6-2- استراتژی جهش

در الگوریتم ژنتیک، جهش (Mutation) نقش حفظ تنوع جمعیت و جلوگیری از گیر افتادن در بهینه‌های محلی را ایفا می‌کند.

## 1-5-6-2 تعویض

جای دو ژن تصادفی در کروموزوم با یکدیگر عوض می‌شود. ساده و سریع است و تغییرات کوچکی ایجاد می‌کند.

## 2-5-6-2- معکوس سازی

یک بازه تصادفی از کروموزوم انتخاب شده و ترتیب ژن‌های آن برعکس می‌شود. ساختار کروموزوم را تا حدی تغییر می‌دهد.

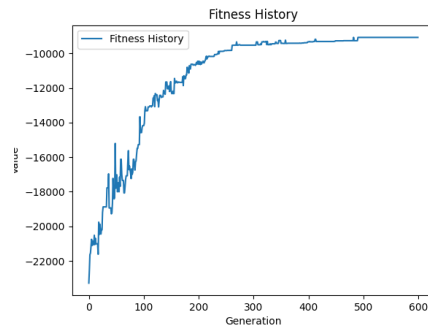
## 2-5-6-3- آشفته‌گی

یک بازه تصادفی از کروموزوم انتخاب شده و ژن‌های آن به‌طور تصادفی بازچینش می‌شوند. بیشترین سطح آشفته‌گی را ایجاد می‌کند.

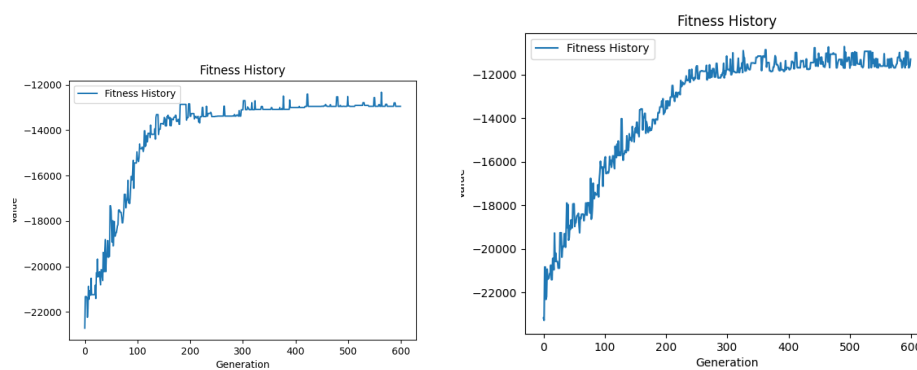
## 2-6-6- مقایسه استراتژی‌های جهش

توضیح تکمیلی سرعت همگرایی تنوع ژنتیکی استراتژی جهش

Swap	کم	زیاد	مناسب برای بهبود جزئی و سریع
Inversion	متوسط	متوسط	ایجاد تعادل بین تنوع و سرعت
Scramble	زیاد		افزایش تنوع بالا، اما کندتر در همگرایی کم



تصویر 19 تعویض



تصویر 20 درهم‌ریختگی

تصویر 21 معکوس سازی

## Abstract

**The Traveling Salesman Problem (TSP)** is a classic optimization problem that aims to find the shortest possible route through a set of cities, visiting each city exactly once and returning to the starting point. Due to its high computational complexity, metaheuristic algorithms such as the **Genetic Algorithm (GA)** are commonly used for solving it. In this project, a genetic algorithm is implemented to solve the TSP on a dataset containing the geographical coordinates of 38 cities in Djibouti.

In this algorithm, chromosomes are carefully designed and parameters such as mutation rate, crossover rate, population size, and number of generations are tuned to balance genetic diversity and convergence speed. Moreover, three selection strategies (tournament, roulette wheel, rank), crossover methods (one-point, two-point, uniform), and mutation strategies (swap, inversion, scramble) are studied and compared to achieve optimal performance.

**Keywords:** Genetic Algorithm, Traveling Salesman Problem, Optimization



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

**Department of Mathematics and Computer Science**

## **Genetic Algorithm**

**By  
AmirMohammadKavosh**

**Supervisor  
Dr.Ghatee**

**Advisor  
Dr.YousefiMehr**

**March 2025**