



# گزارش کار

پردازش تکاملی

پروژه نهایی

آرمین خیاطی

9931153

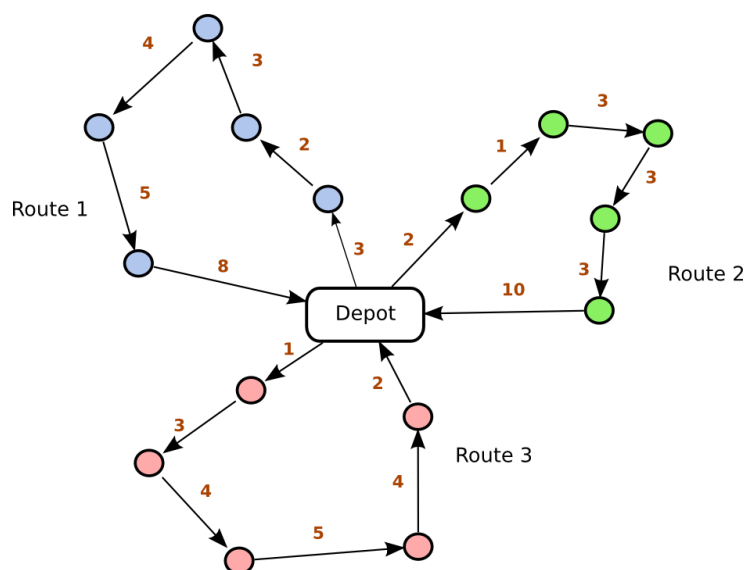
## Contents

3	تشریح مسئله
5	پیاده سازی
5	Coding and Fitness Function
6	عملگر Crossover
6	عملگر Mutation
7	Selection
7	نتایج

## تشریح مسئله

در این پروژه قصد داریم تا مسئله Vehicle Routing را با استفاده از الگوریتم ژنتیک و عملگرهایی که توسط مقاله ارائه شده پیاده سازی کنیم.

صورت مسئله به این صورت است که ما یک مرکز پخش هستیم و تعدادی سفارش از مشتریان خود دریافت کرده ایم. می‌خواهیم به گونه ای سفارش ها را به دست مشتریان برسانیم که با کمترین تعداد خودرو برای حمل بار، سفارش تعداد بیشتری از مشتریان را تحویل دهیم و کوتاه ترین مسیر ها را هم با خودرو های حمل بار طی کنیم. هر مشتری تنها توسط یک خودرو باید سرویس دهی شود و جمع وزن سفارش مشتریانی که یک خودرو می‌خواهد تحویل دهد نباید از ظرفیت آن بیشتر گردد.



طبق این تعریف، حل مسئله به دو نوع تقسیم میشود. در نوع اول تعداد خودرو و یافتن کوتاه ترین مسیر از اهداف مورد نظر است و با استفاده از الگوریتم هایی مانند NSGA و NSGAII آن ها را حل میکنیم. در نوع دوم تعداد خودرو ها ثابت و تنها یافتن بهترین ترکیب از مشتریان برای هر خودرو و کوتاهترین و بهینه ترین مسیر برای خودرو ها ملاک می باشد که از الگوریتم ژنتیک ساده استفاده میشود. مقاله ای که در این پروژه پیاده کرده ایم برای نوع دوم می باشد و از حداقل خودرو های مورد نیاز استفاده میکنیم یعنی حاصل معادله زیر:

$$\text{Minimum Number of Required Trucks} = \left\lceil \frac{\sum_{i=1}^n q_i}{Q} \right\rceil$$

که برابر جمع وزن تعداد کل سفارش ها تقسیم بر ظرفیت خودرو ها میباشد. روابطی که تعاریف ذکر شده را برای این مسئله مدل سازی میکنند به صورت زیر است.

$$\min \quad Z = \sum_{i=0}^N \sum_{j=0}^N \sum_{k=1}^K c_{ij} x_{ijk} \quad (1)$$

$$\sum_{j=0}^N \sum_{k=1}^K x_{ijk} = 1 \quad , i = 1, \dots, N \quad (2)$$

$$\sum_{i=0}^N \sum_{k=1}^K x_{ijk} = 1 \quad , j = 1, \dots, N \quad (3)$$

$$\sum_{i=0}^N \sum_{j=0}^N g_i x_{ijk} \leq Q \quad (4)$$

رابطه اول همان تابع هدف Objective Function است که باید کوتاه ترین مسیر برای هر خودرو را پیدا کند تا سفارشاتش را به مشتریان تحویل دهد. یعنی مجموع مسافت طی شده توسط تمام خودرو ها کمینه باشد. رابطه های دو و سه نیز سرویس دهی شدن مشتریان تنها توسط یک خودرو را بیان میکنند و رابطه چهار محدودیت بیشتر نشدن وزن بار ها از ظرفیت خودرو را نشان میدهد.

```

Initialization
Randomly generate the initial chromosome population
Repeat
k=1
  For p =1:m do
    Choose two chromosomes  $S_i, S_j$  from previous generation
    Produce the chromosome  $S_n$  based on the proposed crossover;
    If  $fit(S_n) > fit(S_i)$ , and  $fit(S_n) > fit(S_j)$ 
      Add  $S_n$  to population
    Else
      Abandon  $S_n$ 
    Excute mutation operation;
  End for
  k=k+ 1
Until (termination conditions)

```

شبه کدی که در بالا میبینید فرایند اجرای الگوریتم ژنتیک ارائه شده برای این مسئله را نشان میدهد.

## پیاده سازی

برای پیاده سازی این مقاله تنها از کتابخانه های استاندارد پایتون، Numpy و Matplotlib استفاده شده است. در ادامه به کد گزاری این مسئله در الگوریتم ژنتیک و عملگر های معرفی شده توسط مقاله میپردازیم.

## Coding and Fitness Function

کد گزاری که برای نمایش یک راه حل در این مسئله استفاده شده، Permutation Representation میباشد. در این کد گزاری هر ژن در کروموزوم بیان گر یک مشتری است یعنی برای 9 مشتری یک راه حل رندم به صورت 5-3-8-2-7-1-6-9-4 می باشد. اگر فرض کنیم برای این مسئله 3 خودرو نیاز میباشد کروموزوم

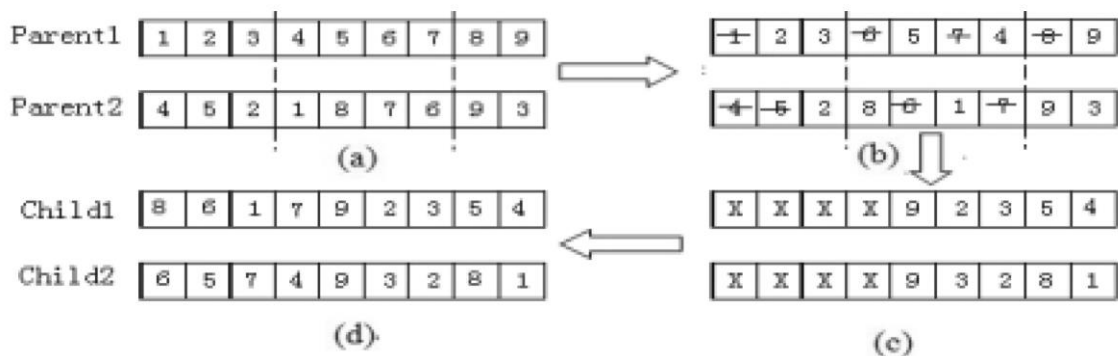
به سه بخش تقسیم میشود 0-2-8-3-5-0, 0-6-1-7-0 و 0-4-9-0 که هر بخش یک مسیر برای یک خودرو برای تحویل سفارشات به مشتری است. از آنجایی که هدف کمینه کردن مجموعه مسیر پیموده شده توسط خودرو ها می باشد، از معکوس این مقدار بعنوان مقدار تابع هدف استفاده میکنیم.

$$fit(Si) = \frac{1}{total\_distance(Si)}$$

## عملگر Crossover

اگر دو والد  $P1 = (123456789)$ ,  $P2 = (452187693)$  را در نظر بگیریم، عملگری که استفاده میکنیم به صورت زیر میباشد.

1. دو نقطه را بصورت رندم انتخاب میکنیم.
2. ژن های بین این دو نقطه را در دو والد به صورت رندم به هم میریزیم.
3. دو زیر رشته را با هم جابجا میکنیم.

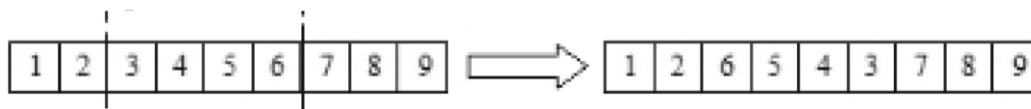


## عملگر Mutation

برای Mutation ما از عملگر Inversion به صورت زیر استفاده میکنیم.

1. دو نقطه را بصورت رندم انتخاب میکنیم.

2. ترتیب ژن های بین دو نقطه را معکوس میکنیم.



## Selection

برای پیاده سازی این مقاله برای Parent Selection از روش Tournament Selection استفاده میکنیم. برای Survival Selection نیز دو بهترین کروموزوم از بین والدین و بچه ها انتخاب و جای والدین قرار میگیرند.

## نتایج

بعد از اجرای الگوریتم با تعداد  $\text{Iteration} = 1000$  مسیر های زیر و نمودار بهترین فیتنس در هر  $\text{Iteration}$  زیر حاصل میشود.

Vehicle 1's route: 0 - 6 - 5 - 3 - 8 - 0

Vehicle 2's route: 0 - 9 - 7 - 4 - 2 - 0

Vehicle 3's route: 0 - 1 - 0

Distance : 553

