



نام و نام خانوادگی: فاطمه حدادی

نام استاد: باقر باباعلی

شماره دانشجویی: ۶۱۰۳۹۶۰۸۷

آذر ۹۹

## گزارش تمرین حل TSP با الگوریتم کلونی مورچه

### • چکیده

در این تمرین سعی شده است که با الهام از الگوریتم کلونی مورچه مسئله TSP یا فروشنده دوره گرد را حل کنیم. پیاده سازی الگوریتم به زبان پایتون و با داده ی bayg29 انجام شده است. همچنین برای بررسی بهتر برازش جواب ساخته شده از روی کلونی در هر مرحله نیز در خروجی برنامه چاپ میشود. در این گزارش به تاثیرگذارترین پارامترها برای اجرای بهتر الگوریتم اشاره میشود.

### • فهرست مطالب

۱. معرفی مسئله و داده ورودی

۲. شرح بخش های تمرین

۳. نمونه خروجی و تحلیل

### ۱. معرفی مسئله و داده ورودی

مسئله TSP یکی از مسائل معروف دسته NP-hard است. این مسئله به صورت داستان یک فروشنده دوره گرد مطرح میشود که قصد دارد یک تعداد شهر را پیماید و به نقطه شروع بازگردد بدون اینکه هر شهر را بیش از یکبار ببیند و کمترین هزینه یا به تعبیری مسافت را داشته باشد. داده ورودی انتخاب شده bayg29 است که به صورت ماتریس هزینه اطلاعات سفر از هر شهر به شهر دیگر ذخیره شده است.

### ۲. شرح بخش های الگوریتم

همانند مراحل الگوریتم در تئوری، ابتدا داده ورودی در ماتریس فاصله ها که همان هزینه است ذخیره میشود که برای محاسبه تابع برازش استفاده میشود. هر مسیر یک جایگشت از شهر ۱ تا ۲۹ است. با حرکت مورچه ها در هر مرحله در مسیرهای عبوری فرمون ریخته میشود و این فرمون ها در آخر جواب بهینه را برای ما میسازند.

برای پیاده سازی الگوریتم از کلاس ACO\_TSP استفاده شده است که دو زیرکلاس Colony و Ant دارد. تابع اصلی در این کلاس run است که طبق کلونی مورچه کلاسیک مسئله را حل میکند. البته برای بهبود عملکرد با روش Max Min ترکیب شده است و در هر مرحله بر روی بهترین مسیر آن مرحله فرمون بیشتری ریخته میشود. در تابع run ابتدا با تابع initializer ابتدا یک مقدار فرمون روی یال ها میریزیم. سپس در یک حلقه به تعداد مراحل انتخاب شده همه ی مورچه ها تور خود را ایجاد میکند و فرمون ها بر اساس برازش هر کدام در تابع updatePheromone دوباره فرمون ریخته میشود البته باید توجه کرد که قبل از ریختن فرمون ابتدا بخشی از

فرمون ها در تابع `evaporatePheromone` با پارامتر تبخیر  $P$  کاسته و تبخیر میشود. بدین ترتیب الگوریتم در بهینه محلی حبس نشده و قابلیت پویش پیدا میکند.

در کلاس `Ant` تابع مهم `maketour` است که طبق فرمون های داخل مسیر در آن مرحله یک تور برروی گراف تشکیل میدهد و برازش آن را با تابع `evaluateTour` محاسبه میشود.

در کلاس `Colony` تابع مهم `showBestTour` است که بر طبق فرمون گذاری یال های گراف بهترین تور را در گراف میسازد.

پارامترهای این مسئله تعداد مورچه های داخل کلونی، تعداد مراحل اجرا، مقدار فرمون اولیه، توان فرمون در محاسبه احتمال انتخاب یک یال، نرخ تبخیر و ضریب افزایش فرمون مسیرها در کلونی است.

در بخش بخشی تلاش میشود هرمدام ازین پارامتر ها را تغییر داده و نتیجه نهایی که بهترین جواب ساخته شده از کلونی در تمام مراحل است را با سرعت الگوریتم بررسی کند.

### ۳. نمونه خروجی و تحلیل

در هر اجرا یک ورودی را تغییر داده ایم که با \* آن را نشان داده ایم.

#### ۱. پارامتر ورودی:

```
ANT_NUMBERS = 29
STEP_NUMBER = 200
INITIAL_PHEROMONE = 10
ALPHA = 1.0
P = 0.1 # evaporate rate
W = 0.1 # pheromone increase rate
```

جواب خروجی:

```
best answer: [0, 7, 26, 22, 6, 24, 18, 10, 13, 16, 21, 17, 19, 9, 3, 14, 2, 28, 25, 8, 4, 5, 27, 11, 20, 1, 12, 23, 15]
fitness: 50.251
```

```
duration: 28.698 s
```

#### ۲. پارامتر ورودی:

```
* ANT_NUMBERS = 50
STEP_NUMBER = 200
INITIAL_PHEROMONE = 10
ALPHA = 1.0
P = 0.2 # evaporate rate
W = 0.1 # pheromone increase rate
```

جواب خروجی:

best answer: [0, 23, 7, 26, 22, 6, 15, 24, 10, 21, 13, 16, 17, 18, 3, 14, 9, 12, 1, 4, 5, 27, 11, 8, 25, 28, 2, 19, 20]

fitness: 50.201

duration: 48.904 s

۳. پارامتر ورودی:

```
ANT_NUMBERS = 29
* STEP_NUMBER = 400
INITIAL_PHEROMONE = 10
ALPHA = 1.0
P = 0.1 # evaporate rate
W = 0.1 # pheromone increase rate
```

جواب خروجی:

best answer: [0, 26, 23, 27, 11, 4, 20, 1, 28, 2, 25, 8, 5, 7, 22, 6, 24, 18, 10, 13, 21, 16, 17, 14, 9, 19, 3, 12, 15]

fitness: 51.76

duration: 53.363 s

۴. پارامتر ورودی:

```
ANT_NUMBERS = 29
STEP_NUMBER = 200
INITIAL_PHEROMONE = 10
* ALPHA = 0.5
P = 0.1 # evaporate rate
W = 0.1 # pheromone increase rate
```

جواب خروجی:

best answer: [0, 7, 26, 22, 6, 24, 18, 10, 13, 16, 21, 17, 19, 9, 3, 14, 2, 28, 25, 8, 4, 5, 27, 11, 20, 1, 12, 23, 15]

fitness: 50.251

duration: 28.698 s

۵. پارامتر ورودی:

```
ANT_NUMBERS = 29
STEP_NUMBER = 200
INITIAL_PHEROMONE = 10
* ALPHA = 0.3
P = 0.1 # evaporate rate
W = 0.1 # pheromone increase rate
```

جواب خروجی:

best answer: [0, 4, 8, 25, 5, 11, 27, 7, 22, 26, 23, 15, 14, 10, 21, 16, 13, 17, 3, 18, 6, 24, 9, 19, 20, 1, 28, 2, 12]

fitness: 47.103

duration: 26.933 s

## ۶. پارامتر ورودی:

```
ANT_NUMBERS = 29
STEP_NUMBER = 200
* INITIAL_PHEROMONE = 20
ALPHA = 1.0
P = 0.1 # evaporate rate
W = 0.1 # pheromone increase rate
```

جواب خروجی:

best answer: [0, 23, 7, 26, 27, 11, 5, 4, 8, 28, 2, 25, 20, 22, 6, 24, 10, 21, 16, 13, 14, 18, 15, 12, 3, 19, 9, 17, 1]  
fitness: 47.438

duration: 26.618 s

## ۷. پارامتر ورودی:

```
ANT_NUMBERS = 29
STEP_NUMBER = 200
INITIAL_PHEROMONE = 10
ALPHA = 1.0
* P = 0.9 # evaporate rate
W = 0.1 # pheromone increase rate
```

جواب خروجی:

best answer: [0, 11, 5, 4, 8, 25, 28, 2, 3, 21, 16, 13, 17, 14, 10, 24, 6, 22, 7, 23, 26, 20, 1, 19, 9, 18, 15, 12, 27]  
fitness: 49.334

duration: 26.072 s

## ۸. پارامتر ورودی:

```
ANT_NUMBERS = 29
STEP_NUMBER = 200
INITIAL_PHEROMONE = 10
ALPHA = 1.0
P = 0.1 # evaporate rate
* W = 0.9 # pheromone increase rate
```

جواب خروجی:

best answer: [0, 9, 19, 15, 23, 12, 1, 3, 18, 24, 21, 16, 17, 13, 14, 10, 6, 22, 2, 28, 4, 8, 25, 20, 5, 27, 26, 7, 11]  
fitness: 40.032

duration: 25.451 s

## • تحلیل

ابتدا لازم به ذکر است که نتیجه گیری های پیش رو صرفا با همین ۸ نمونه ی اجرا به دست نیامده و هر تغییر متغیر به ازای ورودی های مختلف امتحان شده است. همچنین در الگوریتم کلونی مورچه جواب

های به دست آمده با ورودی یکسان در اجراهای مختلف متفاوت است پس بهتر است روی روند افزایش و کاهش به همراه مشاهدات استدلال کرد.

اجرای اول یک اجرا ی سبک و ساده از الگوریتم است. طبق نظر محققین بهتر است تعداد مورچه ها از تعداد گره های گراف کمتر نباشد. برای همین ۲۹ مورچه در هر مرحله بر روی کلونی حرکت میکنند. طبق اجرای دوم با افزایش مورچه ها البته نتیجه چشم گیری به دست نمی آید. فقط گاهی برازش آخر تا حدی افزایش پیدا میکند. همچنین همانطور که مشاهده میشود سرعت رسیدن به جواب الگوریتم افزایش یافته است.

باید توجه کرد با افزایش مورچه ها فرمون های بیشتری در مسیر ریخته میشود که یا سبب بهینه محلی زود هنگام شده یا ورن یال ها را به هم نزدیک میکند.

در اجرای سوم تعداد مراحل افزایش یافته که سبب بهبود نتیجه نهایی شد اما سرعت را به شدت کاهش داد!

در اجرای چهارم و پنجم پارامتر آلفا افزایش یافته است اما این پارامتر طی آزمایش ها مشاهده شد تغییر زیادی را سبب نمیشود و برای فرمون هایی با مقدار بزرگ مناسب هستند. اما خوش بختانه بر روی سرعت بی تاثیر هستند.

در اجرای ششم مقدار فرمون در ابتدای الگوریتم افزایش یافته است. اما باید توجه داشت مقدار فرمون باید متناسب با فرمون ها در مراحل بعدی بودی و نسبت به آن مراحل حداقل ممکن باشد. پس هم مقدار فرمون کم و هم مقدار فرمون زیاد کمک کننده نیستند.

در اجرای شماره هفتم پارامتر P افزایش یافته که این کار از افتادن در بهینه محلی جلوگیری کرده و قابلیت پویا را افزایش میدهد. اما باید توجه کرد در مواردی با نرخ تبخیر بالا قدرت انتفاع به شدت کاهش یافته و جواب آخر مطلوب ما نیست. البته مانند قبلی تاثیر به ویژه ای در سرعت اجرای الگوریتم ندارد.

در اجرای آخر مقدار وزن برای افزایش فرمون مسیر هر مورچه افزایش یافته اما با این کار همانطور که قابل مشاهده است ما را به جواب بهینه نزدیک نکرد زیرا از قدرت فرمون گذاری بهترین تور مورچه در آن گراف می کاهد. البته این روش با کاستن از بار محاسباتی در کل کمی به سرعت می افزاید.

در کل درباره ی این الگوریتم باید به نحوه مقدار دهی به پارامترها به شدت دقت شود زیرا بسیار حساس است. اما در صورت پارامتر گذاری مناسب این الگوریتم جواب خوب و سریعی را نسبت به سایر ورژن های کلونی مورچه بدون استفاده از پارامتر هیوریستیکال به ما میدهد که نشان از قوی بودن آن دارد.

هدف کلی من از انتخاب این نوع از الگوریتم کلونی مورچه در ابتدا شک من به کاربردی بودن نوع ساده ی آن بود که تنها با فرمون ها مسیر را انتخاب میکند. اما در آخر توانستم ببینم حتی بدون کمک فرمون گذاری بهترین جواب الگوریتم باز هم همگرا شده فقط سرعت همگرایی پایین تر است.

## سپاس گذاری

در آخر از استاد گرامی دکتر باقر باباعلی تشکر میکنم.