

بهینه‌سازی زنجیره تأمین با استفاده از الگوریتم کلونی مورچگان

امیرحسین قناعتیان
۹۷۱۰۴۵۸۳

استاد پروژه:
جناب دکتر حسن ناییبی

شهریور ۱۴۰۱



فهرست



✓ الگوریتم کلونی مورچگان

✓ تشریح مسئله

✓ پیاده‌سازی الگوریتم روی داده‌ها

✓ نتایج

منشأ الهام الگوریتم

مورچه‌ها در ضمن حرکت خود به سمت منبع غذایی، ردی از فرومون در محیط منتشر می‌کنند که به‌طور طبیعی و با گذر زمان متلاشی می‌شود.

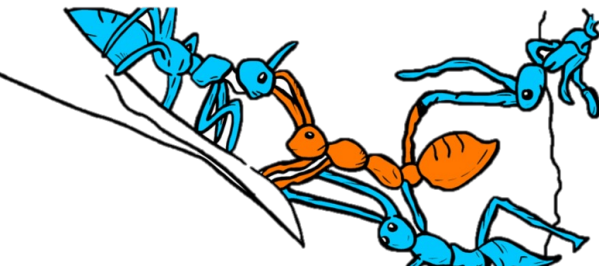
مورچه‌ای که کوتاهترین مسیر به سمت منبع غذایی را انتخاب کرده، سفر برگشتی به سمت آشیانه را زودتر از دیگر مورچه‌ها آغاز می‌کند.

در چنین حالتی، این مورچه در مسیر بازگشت به آشیانه، دوباره شروع به منتشر کردن فرومون در محیط می‌کند و از این طریق، رد فرومون به جا گذاشته در کوتاهترین مسیر را تقویت می‌کند.

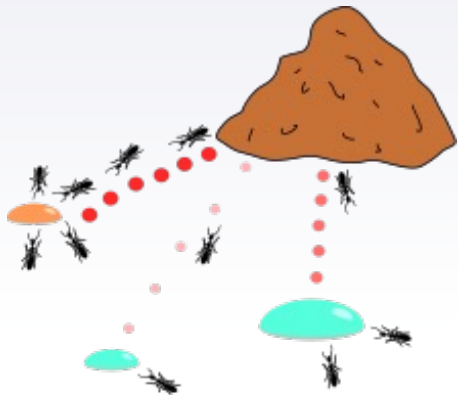
مورچه‌های دیگر، به‌طور غریزی، قوی‌ترین مسیر فرومون موجود در محیط را دنبال و رد فرومون در این مسیر را تقویت می‌کنند.

پس از گذشت مدت زمان مشخصی، نه تنها رد فرومون موجود در کوتاهترین مسیر متلاشی نمی‌شود، بلکه، با انباشته شدن رد فرومون دیگر مورچه‌ها، بیش از پیش تقویت می‌شود.

مسیری که قوی‌ترین رد فرومون در آن به جا گذاشته شده باشد، به مسیر پیش فرض برای حرکت مورچه‌ها از کلونی به منبع غذایی و برعکس تبدیل می‌شود.



نحوه استفاده از الگوریتم

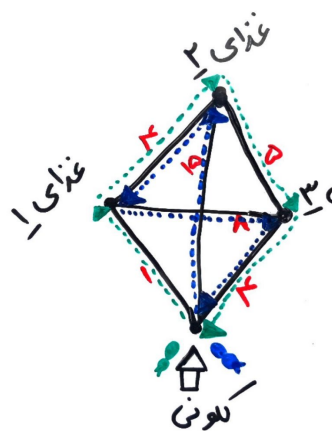
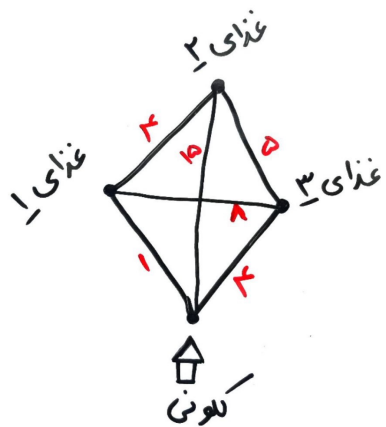


محققان، در پی مشاهده نتایج آزمایش پل باینری، در صدد آن برآمدند تا مدلی ریاضی را برای توصیف رفتار بهینه مورچگان توسعه دهند. با فرض اینکه پس از گذشتن t واحد زمانی از زمان آغاز آزمایش، m_1 مورچه از پل اول و m_2 مورچه از پل دوم استفاده کرده باشند، احتمال (p_1) اینکه مورچه $m+1$ پل اول را انتخاب کند در معادله زیر نشان داده شده است.

$$p_1(m+1) = \frac{(m_1 + k)^h}{((m_1 + k)^h + (m_2 + k)^h)}$$

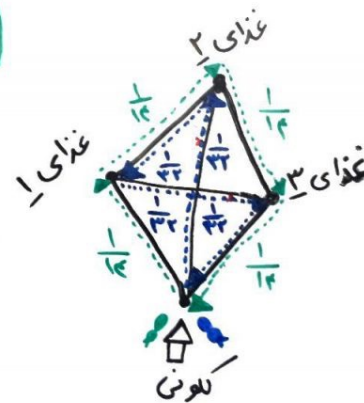
در این رابطه، پارامترهای h و k ، پارامترهای مورد نیاز برای برازش مدل روی داده‌های آزمایش است. همچنین، احتمال اینکه همان مورچه پل دوم را انتخاب کند، از طریق معادله $p_2 = 1 - p_1$ به دست می‌آید. شبیه‌سازی مونت کارلو انجام شده روی داده‌های آزمایش نشان می‌دهد که مدل توسعه داده شده با مقادیر پارامترهای $h \approx 2$ ، $k \approx 20$ ، برازش بسیار خوب و مناسبی از داده‌های آزمایش تولید می‌کند. از این مدل ساده می‌توان برای طراحی مورچه‌های مصنوعی استفاده کرد که مسائل بهینه‌سازی را به شیوه مشابه حل می‌کنند.

یک مثال ساده

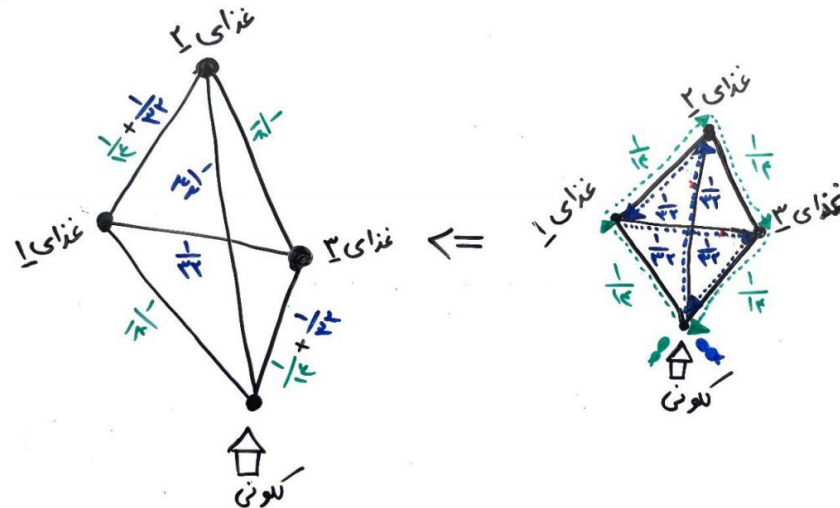


مورچه سبز $\Rightarrow 1 + 4 + 5 + 4 = 14 \rightarrow \frac{1}{14}$

مورچه آبی $\Rightarrow 15 + 4 + 8 + 4 = 32 \rightarrow \frac{1}{32}$

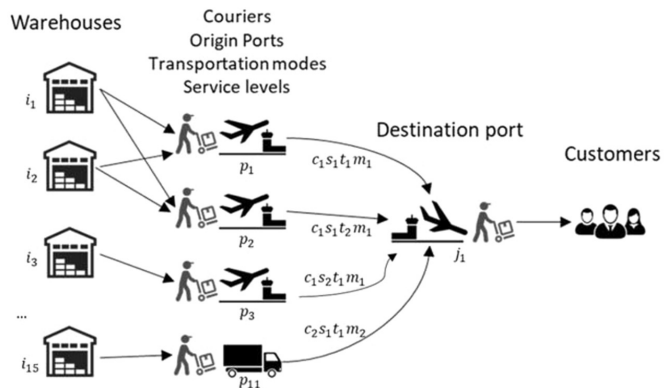


نتیجه‌گیری مورچه در مرحله بعد، احتمالی است که با استفاده از فرومون موجود به دست می‌آید. بیشترین فرومون در مسیر غذای شماره ۳ است، پس در تکرار بعدی احتمال گذر از این مسیر از سایر مسیرهای موجود بیشتر خواهد بود.

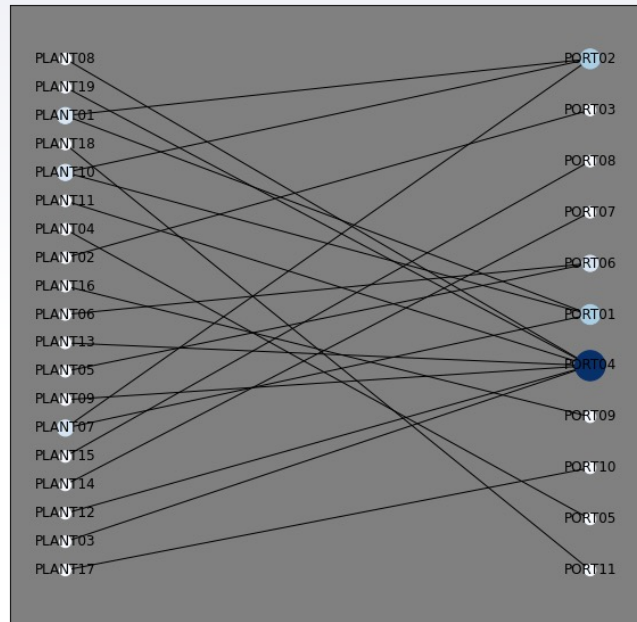


تشریح مسئله

محصولات تولیدی
انبارهای ذخیره سازی
مبدا اولیه
نحوه حمل و نقل
سطح سرویس دهی



محدودیت‌های مسئله



$$\sum_{k=1}^l o_{ki} \leq C_i \quad \sum_{k=1}^l w_{kpjcostm} \leq \max F_{pjcostm} \quad k_z \in i_z$$



تابع هدف

هزینه انبار $WC_{k,i}$ و هزینه حمل و نقل $TC_{k,p,j}$

WC هزینه انبار برای سفارش k در انبار i و TC هزینه حمل و نقل برای سفارش k بین انبار p و مشتری j

s_k سطح خدمت برای سفارش p ، k پورت مبدا، j پورت مقصد، نوع پیک، s سطح خدمت، t زمان تحویل،
 m حالت حمل و نقل است. علاوه بر این، $M_{(p,j,c,s,t,m)}$ حداقل هزینه برای مسیر مشخص شده است،
 $w_{(k,p,j,c,s,t,m)}$ وزن بر حسب کیلوگرم برای سفارش k است و $R_{(p,j,c,s,t,m)}$ نرخ حمل و نقل (دلار
 برای هر کیلوگرم) برای وزن های معین بر اساس وزن کل خط انتقال مطابق با جدول نرخ حمل و نقل است.

$$\min \sum_{k=1}^I (WC_{ki} + TC_{kpj})$$

$$WC_{ki} = q_k \times P_i$$

Transportation cost (TC_{kpj})

1. if $s_k = CRF$
2. then $TC_{kpj} = 0$
3. else if $m = GROUND$
4. then $TC_{kpj} = \frac{R_{pjcostm}}{\sum_{k=1}^I w_{kpjcostm}} \times w_{kpjcostm}$
5. else
6. then $TC_{kpj} = R_{pjcostm} \times w_{kpjcostm}$
7. if $TC_{kpj} < M_{pjcostm}$
8. then $TC_{kpj} = M_{pjcostm}$
9. end if
10. end if

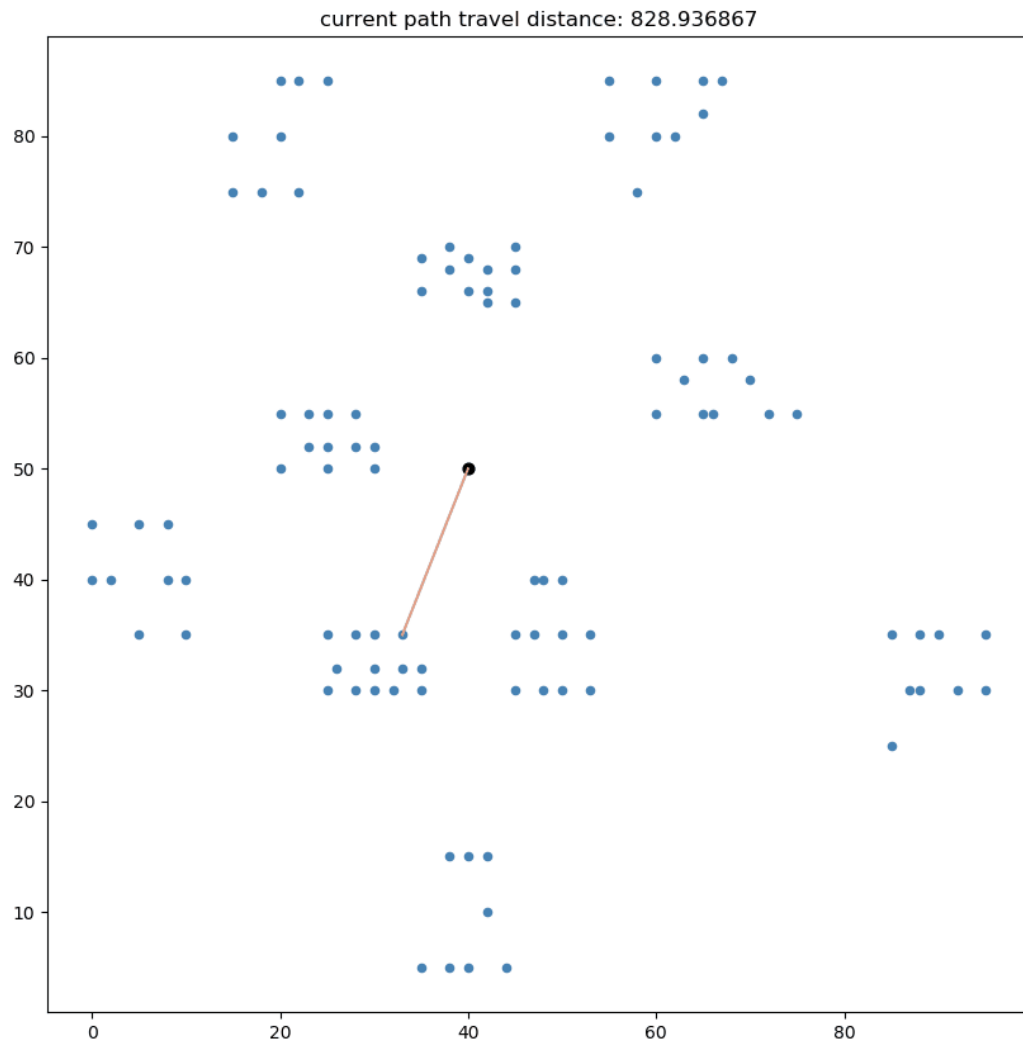


پیاده‌سازی الگوریتم

- ✓ وارد کردن داده‌ها به برنامه
- ✓ تخصیص دیکشنری یا لیست مناسب به داده‌ها
- ✓ فرایند پیش‌پردازش داده‌ها
- ✓ تعریف توابعی برای محدودیت‌ها و تابع هدف
- ✓ تعریف کلاسی از مورچگان، کلونی، گره‌های گراف (مبدأ و مقصد)
- ✓ اجرای برنامه با تکرار ۱۰۰

```
parameters = {  
    'swarm_size': 250,  
    'min_values': (-5, -5),  
    'max_values': (5, 5),  
    'iterations': 100,  
    'decay': 0,  
    'w': 0.9,  
    'c1': 2,  
    'c2': 2  
}
```

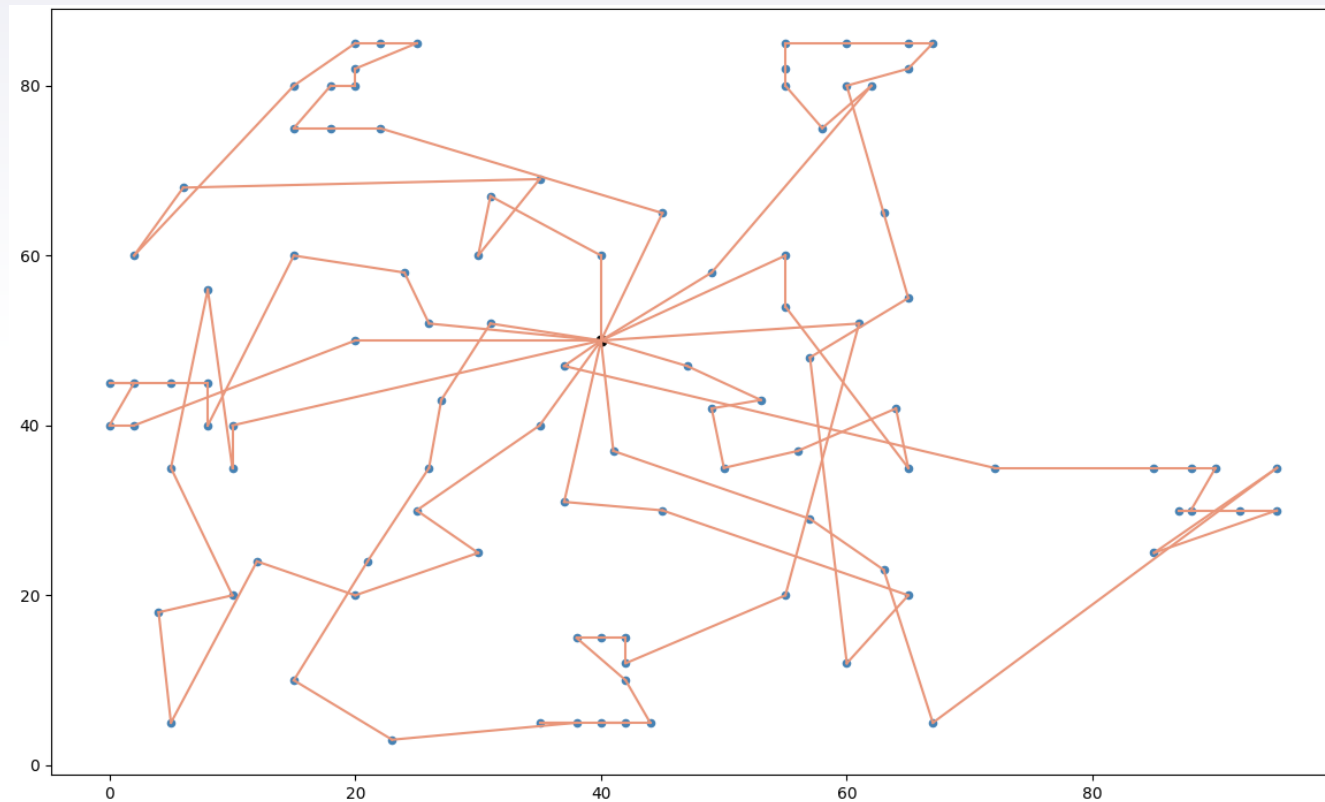




یک تکرار از الگوریتم

یک جواب بهینه، نتیجه‌ی ۱۰۰ تکرار برنامه

12



جزئیات بخشی از یک اجرا با ۱۰۰ تکرار

[iteration 2]: find a improved path, its distance is 1805.362395
it takes 0.733 second multiple_ant_colony_system running

[iteration 3]: find a improved path, its distance is 1717.261830
it takes 0.883 second multiple_ant_colony_system running

[iteration 4]: find a improved path, its distance is 1413.867002
it takes 1.062 second multiple_ant_colony_system running

[iteration 12]: find a improved path, its distance is 1407.822424
it takes 2.550 second multiple_ant_colony_system running

[iteration 26]: find a improved path, its distance is 1290.919178
it takes 5.117 second multiple_ant_colony_system running

[iteration 36]: find a improved path, its distance is 1285.391084
it takes 7.082 second multiple_ant_colony_system running

[iteration 52]: find a improved path, its distance is 1225.292652
it takes 11.209 second multiple_ant_colony_system running

[iteration 91]: find a improved path, its distance is 1219.253686
it takes 16.336 second multiple_ant_colony_system running



سیاست بهینه

✓ بهترین انبار برای ذخیره سازی (نگهداری)

✓ هزینه ی انبارداری

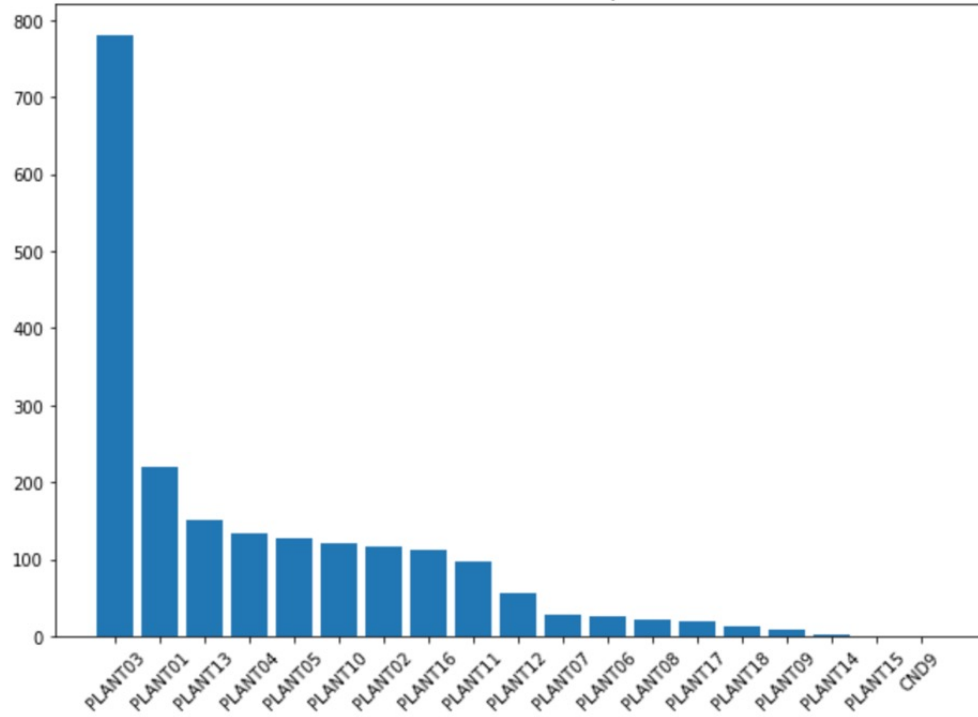
✓ بهترین مقصد اولیه

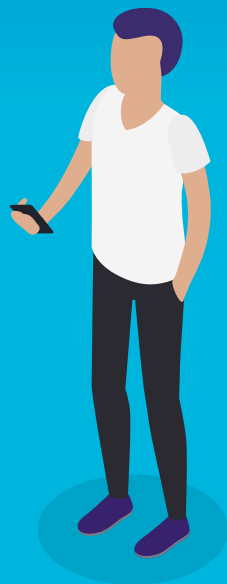
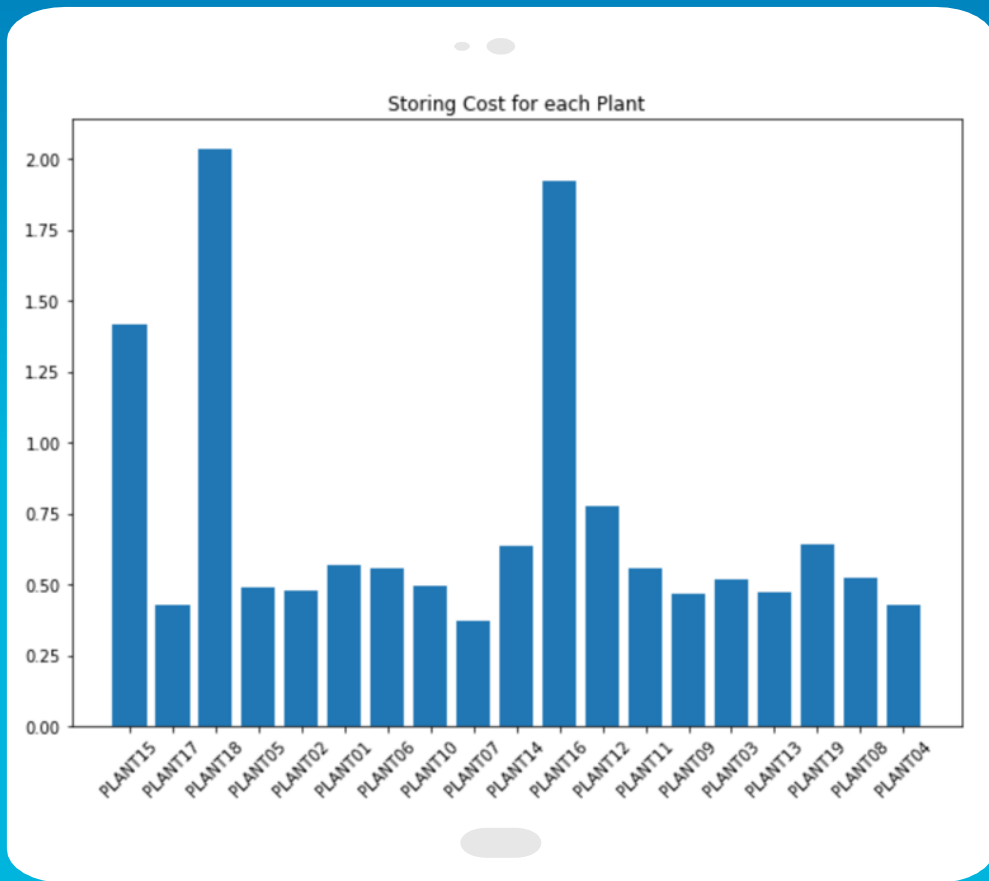
✓ هزینه ی انتقال

	min_cost	best_plant	best_port_price	best_port
0	4.756374	PLANT16	2.836567	PORT09
1	4.756374	PLANT16	2.836567	PORT09
2	4.756374	PLANT16	2.836567	PORT09
3	4.756374	PLANT16	2.836567	PORT09
4	4.756374	PLANT16	2.836567	PORT09



Number of Products that each plant Stores





با تشکر از توجه شما.

