

حل توابع با روش ازدحام ذرات

فایل کد در PSO.py قرار دارد.

جواب ۱۹.۲۱- برای تابع اول و ۰.۲۹ برای تابع دوم توسط کد من بدست آمد.

توضیحات الگوریتم

در ابتدا نقاط را به صورت تصادفی در بازه داده شده به اندازه count یا همان تعداد اولیه قرار میدهم و در هر مرحله سرعت و مکان ها تغییر میکنند.

```
FindMin(count, function1, -10, 10, w, c1, c2, end_point)
```

- **سرعت:** سرعت اولیه به صورت تصادفی در بازه ۱ تا ۳ داده می شود و هربار سرعت به صورت زیر به روز رسانی میشود.

$$v(t+1) = w * v(t) + c1 * \text{random}(-1,1) * (Pbest - x(t)) + c2 * \text{random}(-1,1) * (Gbest - x(t))$$

- **مکان نقاط:** در هر مرحله $x(t+1) = x(t) + v(t)$ خواهد شد.
- **بهترین خود (Pbest):** در هر مرحله مقدار کنونی تابع با مقدار های قبلی آن بررسی میشود و کمترین آن در Pbest ذخیره میشود.

- **بهترین کل جمعیت (Gbest):** در هر مرحله تمام نقاط بررسی میشوند و بهترین آن (یعنی آنکه تابعش جواب کمتری دارد) نسبت به Gbest قبلی جایگزین میشود.

- **پایان الگوریتم:** الگوریتم بعد از آنکه بهترین جواب (Gbest) بعد از end_point مرحله تغییر نکرد متوقف میشود.

تحلیل پاسخ های بدست آمده

تقریباً در تمام مواقع اگر ما پارامترها را به صورت زیر قرار دهیم جواب ها همیشه یکسان و برابر مقدار گفته شده در بالا هستند.

```
count = 1000
w = 0.5
c1 = 0.5
```

```
c2 = 0.5  
end_point = 100
```

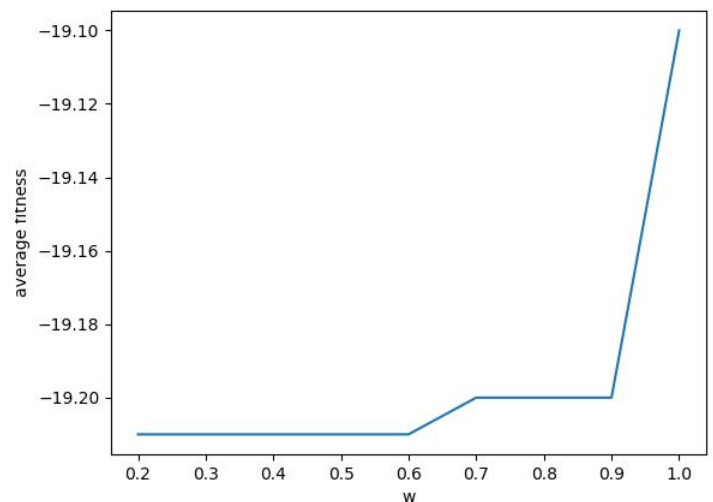
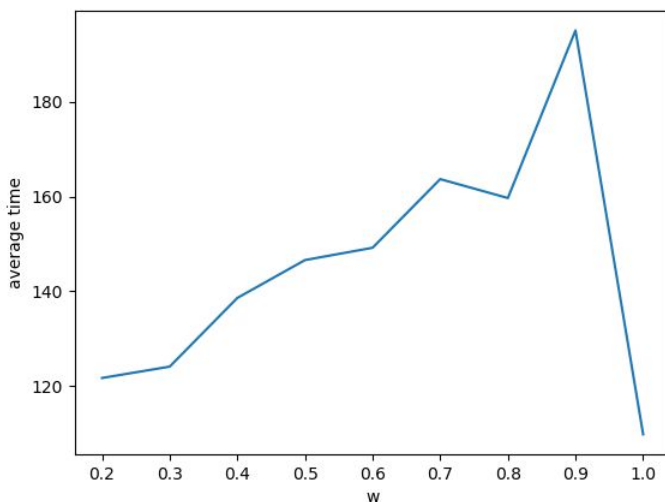
نقطه ها هیچ وقت ثابت نمی مانند اما همیشه بهترین جواب بدست می آید و یکتاست.

تاثیرات پارامترها

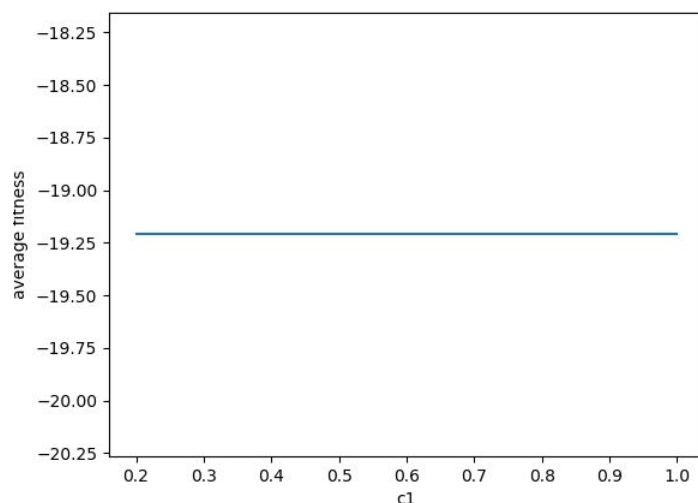
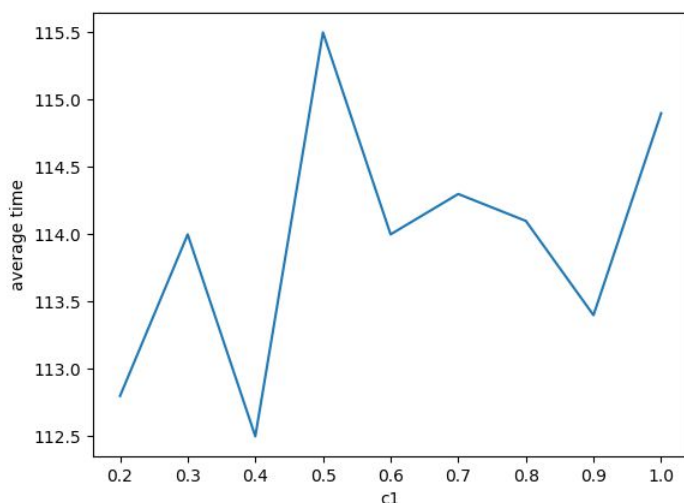
هر برنامه با پارامترهای مشخص شده ۱۰ بار اجرا شده اند.

تابع اول

- **تغییر w:** طبق نمودارها با افزایش آن تا ۰.۹ تاثیری بدی دارد ولی در ۱ زمان بهتری دارد ولی میانگین جواب های بدست آمده خوب نیستند. پس تا ۰.۹ می توان از آن نتیجه خوبی گرفت.



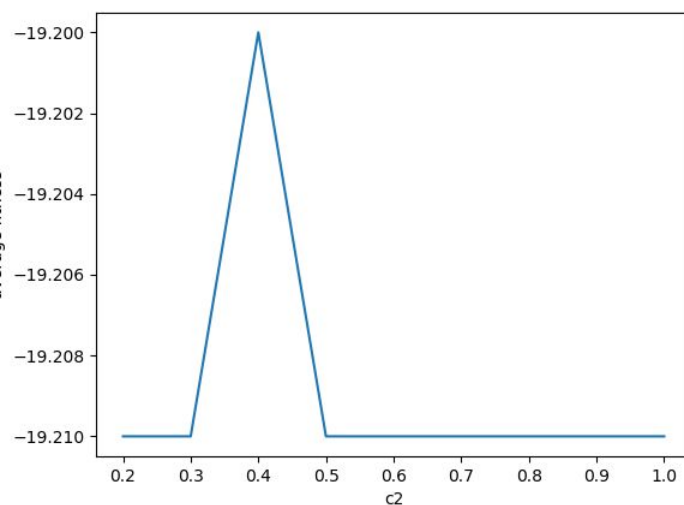
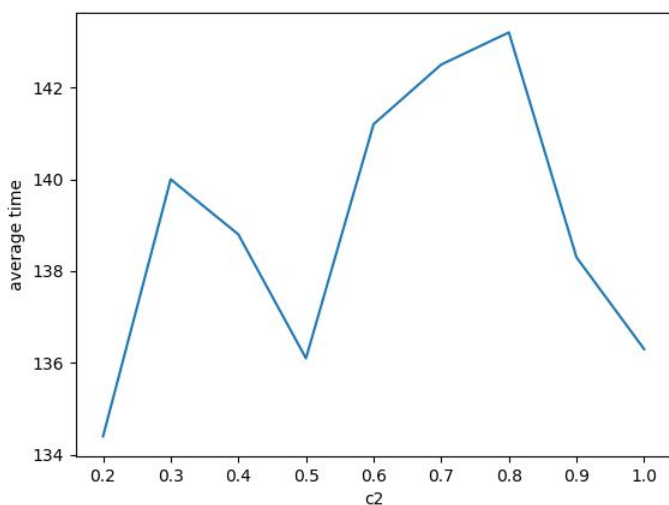
- **تغییر c1:** تغییر این پارامتر بسیار رفتار تصادفی ای در زمان دارد و طبق نمودار در ۰.۴ بهترین تاثیر را دارد چون زمان خیلی کمی صرف میکند. طبق نمودار سمت راست تغییر این پارامتر در جواب های مسئله تغییری ایجاد نمی کند. بطور کلی با کاهش این پارامتر رفتار بهتری دارد.



● **تغییر c2:** تغییر این پارامتر نیز رفتار تصادفی دارد اما در کمترین مقدار یعنی ۰.۲ و بیشترین مقدار یعنی

۱ کمترین زمان را صرف میکند. در ۰.۴ میانگین جواب ها اصلا خوب نیست. بطور کلی جواب ها با

کاهش این پارامتر رفتار بهتری دارند.

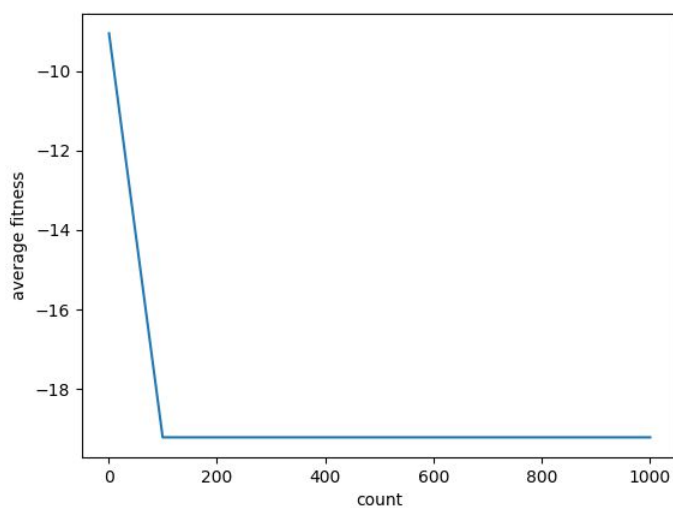
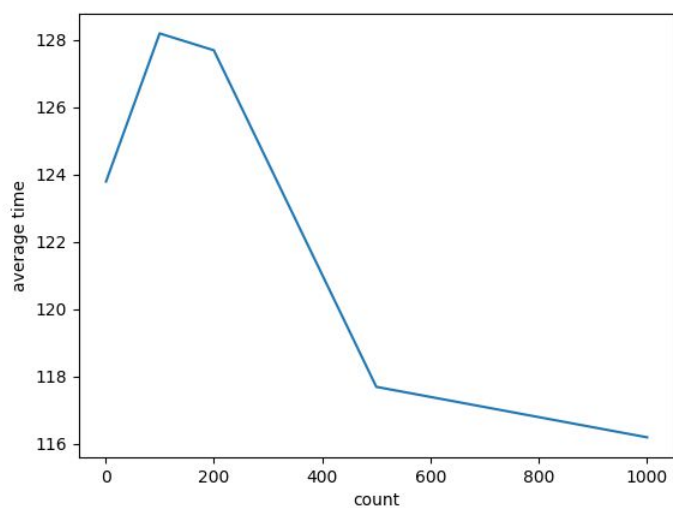


● **تغییر تعداد نقاط اولیه (count):** اگر یک نقطه در صفحه قرار دهیم نه تنها جواب درستی بدست نمی

آورد بلکه زمان زیادی نیز صرف میکند. ولی در بقیه موارد واضح است که تعداد نقاط بیشتر هم زمان

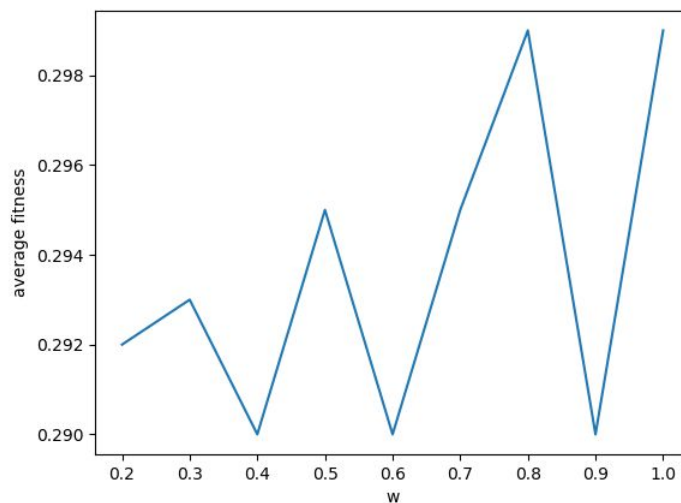
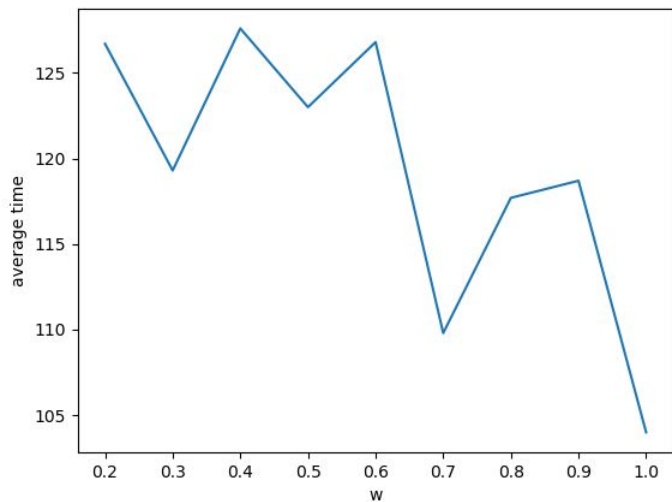
کمتری نیاز دارند و هم جواب بهینه را همیشه بدست می آورند. بطور کلی افزایش تعداد نقاط بسیار عالی

است.

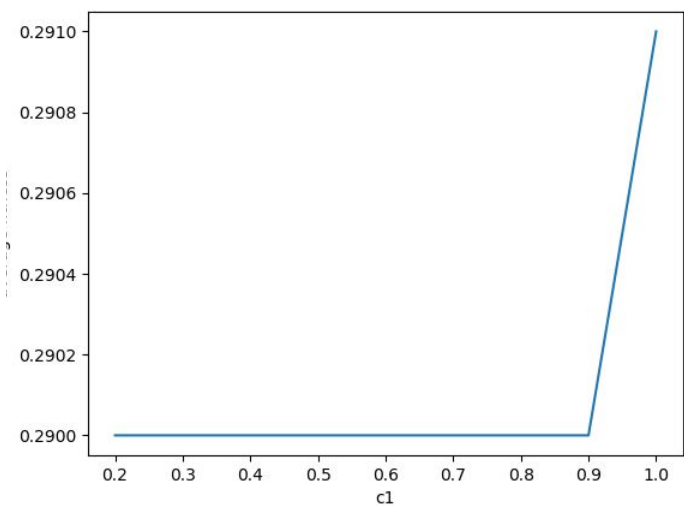
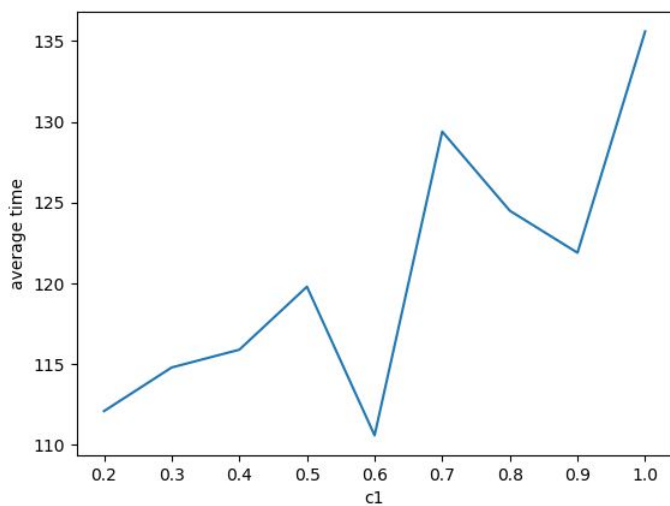


تابع دوم

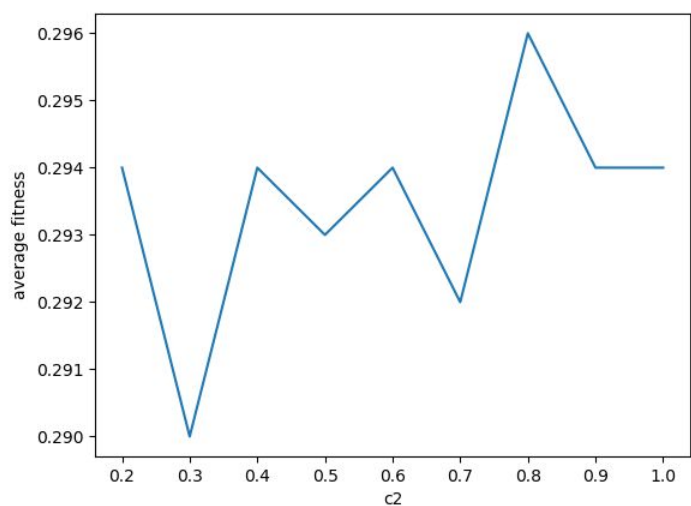
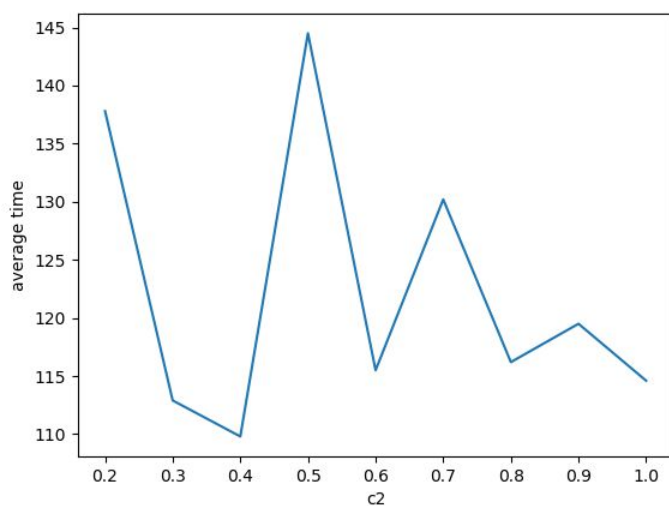
● **تغییر w :** طبق نمودارها افزایش w زمان را بسیار کاهش میدهد و جواب را تا حدودی خوبی بهینه بدست میآورد. پس بطور کلی افزایش w کارآمد است.



● **تغییر $c1$:** طبق نمودار افزایش این پارامتر هم در زمان بسیار بد است و هم در بدست آوردن کمینه با جواب های بقیه اندکی فرق دارد. بطور کلی کاهش این پارامتر بهتر است.



- **تغییر c2:** طبق نمودار رفتار این واریامتر کاملاً تصادفی است. در مورد زمان کاملاً در حال تغییر است و میانگین جواب‌ها بسیار نزدیکند. بطور کلی در نزدیک ۱ و ۰.۴ زمان‌های کمتری مصرف می‌کند و میتوان گفت افزایش آن خوب است.



- **تغییر تعداد نقاط اولیه (count):** طبق نمودار تا ۱۰۰ تا نقطه بسیار زمان و میانگین کمی دارد اما از ۲۰۰ به بعد هرچه بیشتر باشد بسیار بهتر عمل می‌کند.

