

توضیحات کد

ستاره روشن

حسین مختاریان

محمدرضا رضائی

است. به یاد داشته باشید که این ابزارها در خود مسئله استفاده می‌شوند نه در مسئله بهینه سازی مورد بحث ما.

```
1 toolbox.register("evaluate", evaluation)
2 toolbox.register("cx_pm", tools.cxPartiallyMatched)
3 toolbox.register("cx_onepoint", tools.cxOnePoint)
4 toolbox.register("cx_twopoint", tools.cxTwoPoint)
5 toolbox.register("mut_shuffle", tools.mutShuffleIndexes)
6 toolbox.register("mut_inverse", mutInv)
7 # indpb = random.randint(0,10)/10
8 #toolbox.register("select", tools.selTournament, tournsize = 10)
9 toolbox.register("sel_roulette", tools.selRoulette)
10 toolbox.register("sel_best", tools.selBest)
11 toolbox.register("sel_worst", tools.selWorst)
12 toolbox.register("sel_random", tools.selRandom)
```

شکل ت. تعریف تمامی متدها. لازم به ذکر است **tounment** در قسمت **main** در بدنه کد تعریف شده است (به علت امکان تغییر ساین).

2. انتخاب

در این قسمت ما بهترین پارامترها را برای الگوریتم انتخاب کردیم. خط دوم در شکل ث ایندکس اعضای آرایه را بصورت نزولی برمیگرداند پس از آخر ما ۲۰ تا از بهترین‌ها را انتخاب کردیم.

```
1
2 arg = np.argsort(mbf)
3 new = np.zeros(140).reshape(20,7)
4
5 for i in range(20):
6     new[i,:] = Meta_GA[arg[19-i],:]
7 new
```

شکل ث. انتخاب بهترین الگوریتم.

3. Crossover

در آخر فقط از crossover استفاده شده و از جهش استفاده نشده. جمعیت آخر هم به تعداد 20 عدد انتخاب کردیم. در قسمت crossover مکانی نا مشخص در کروموزوم انتخاب خواهد شد و از آن قسمت جابجایی خواهیم داشت.

```
1 def cx(k1, k2):
2     rand = random.randint(0,6)
3     for i in range(rand, 3):
4         temp = k1[i]
5         k1[i] = k2[i]
6         k2[i] = temp
7     return k1,k2
8
9 final = np.zeros(140).reshape(20,7)
10 for i in range(10):
11     index1 = random.randint(0,3)
12     index2 = random.randint(0,3)
13     final[i,:], final[19-i,:] = cx(new[index1,:], new[index2,:])
14
15 mbf_final = np.zeros(20)
16 for i in range(20):
17     res1, win = main (Selectlist[int(final[i,4])], CXlist[int(final[i,5])],mut[int(final[i,6])], final[i,0], final[i,1],
18     # res1, win = main (Selectlist[int(Meta_GA[i,4])], CXlist[int(Meta_GA[i,5])],mut[int(Meta_GA[i,6])])
19     # ,Meta_GA[i,0], Meta_GA[i,1], int(Meta_GA[i,2]), int(Meta_GA[i,3]))
20     mbf_final[i] = np.mean (res1)
21 # plt.plot (res1, label = 'new data')
```

شکل ج crossover

چکیده — در این گزارش به بررسی کد مربوط به بهینه سازی الگوریتم تکاملی پرداخته خواهد شد.

1. جمعیت اولیه

در درس پارامترها به دو دسته تقسیم شده‌اند عددی و غیر عددی در این پیاده سازی ما هر دو را مورد بررسی قرار دادیم. کروموزوم مورد استفاده:

Mutation probability	Crossover probability	Tournament size	Population size	Selection method	Crossover method	Mutation method
----------------------	-----------------------	-----------------	-----------------	------------------	------------------	-----------------

```
1 mbf = np.zeros(50)
2 Meta_GA = np.zeros(350).reshape(50,7)
3 for i in range(50):
4     Meta_GA[i,1] = random.randint(0,10)/10 #crossover
5     Meta_GA[i,0] = random.randint(0,10)/10 #mutation
6     Meta_GA[i,2] = random.randint(1,100) #tournament size
7     Meta_GA[i,3] = random.randint(100,200) #population
8     Meta_GA[i,4] = random.randint(0,4) #selection
9     Meta_GA[i,5] = random.randint(0,2) #crossover
10    Meta_GA[i,6] = random.randint(0,1) #mutation
11    res, win = main (Selectlist[int(Meta_GA[i,4])], CXlist[int(Meta_GA[i,5])],mut[int(Meta_GA[i,6])])
12    ,Meta_GA[i,0], Meta_GA[i,1], int(Meta_GA[i,2]), int(Meta_GA[i,3]))
13    #main (selection, CX, mutation, CXPS, MUTPS, tszie, popul)
14    mbf[i] = np.mean(res, axis = None)
```

شکل أ. تعریف جمعیت اولیه

حال با مشاهده شکل بالا؛ خط ۴ احتمال crossover خط ۵، احتمال mutation خط ۶ ساینز تورنمنت (در صورت استفاده و در غیر این صورت نادیده گرفته خواهد شد)، خط هفت اندازه جمعیت الگوریتم اصلی و خطوط ۸، ۹ و ۱۰ به ترتیب نوع انتخاب، crossover، و جهش است (شکل ب).

به منظور استفاده ساده‌تر تمامی انواع انتخاب، crossover همچنین جهش تعریف شده است از قبل که تنها نیاز به فراخوانی خواهند داشت (شکل ت). ما ۵۰ تا جمعیت اولیه به صورت رندوم انتخاب نمودیم و سپس best fitness mean هر کدام را در خط ۱۴ شکل أ بدست آوردیم.

```
1 Selectlist = ['trnmnt', 'roulette', 'best', 'worst', 'random']
2 CXlist = ['onepoint', 'twopoint', 'pmx']
3 mut = ['shuffle', 'inverse']
```

شکل ب. آرایه‌های تعریف شده به منظور استفاده از به ترتیب selection که به ترتیب انواع tournament, roulette, select best, select worst و انتخاب رندوم میباشد. خط ۲ انواع crossover که شامل تک نقطه، دو نقطه و pmx است و سپس mutation در خط سوم که shuffle و inverse