

# تمرین سری چهارم الگوریتم‌های مدرن در بهینه‌سازی

علی بنی‌اسد

۲۳ دی ۱۴۰۱

## ۱ سوال اول

برای بهینه‌سازی از الگوریتم PSO استفاده شده است. در تابع پیاده سازی شده ابتدا تعدادی پرنده به صورت تصادفی در بازه تعریف شده قرار می‌گیرند. برای هر پرنده یک سرعت اولیه به صورت تصادفی انتخاب می‌شود. تغییرات سرعت پرنده به صورت معادله ۱ است.

$$v = wv + C_1 N(0, 1)(p_{best} - x) + C_2 N(0, 1)(g_{best} - x) \quad (1)$$

$$x = x + v \quad (2)$$

در رابطه بالا ضرایب  $w$ ،  $C_1$  و  $C_2$  از ضرایب رایج در مقالات استفاده شده است.  $p_{best}$  بیانگر بهترین تجربه هر پرنده است و  $g_{best}$  بیانگر بهترین تجربه‌ی پرندehایی است که با آن در ارتباط است. توپولوژی همسایگی (شکل ۱) به صورت حلقه (Ring) در نظر گرفته شده است. در این سوال خواسته بخش اول گزارش CEC2005 انجام شده است (با توجه به اینکه حجم محاسبات بالایی داشت و ددلاین نزدیک بود دیتاهای بدست آمده در جدول آورده شده است و سایر دیتاها به محض بدست آمدن اضافه خواهند شد و در github به روزرسانی خواهد شد)، ولی، در گزارش تنها بخشی از نمودارها آورده شده است و سایر نمودارها در فایل Figure و بخش ۳ است. پارامترهای الگوریتم PSO در جدول ۱ آورده شده است.

شکل ۱: توپولوژی همسایگی حلقه

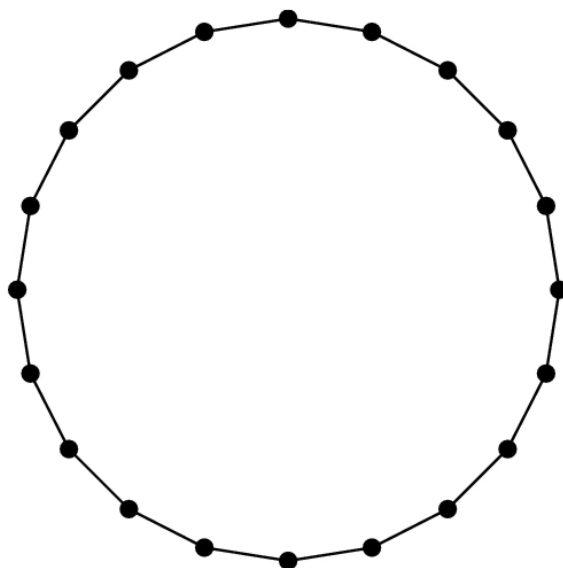
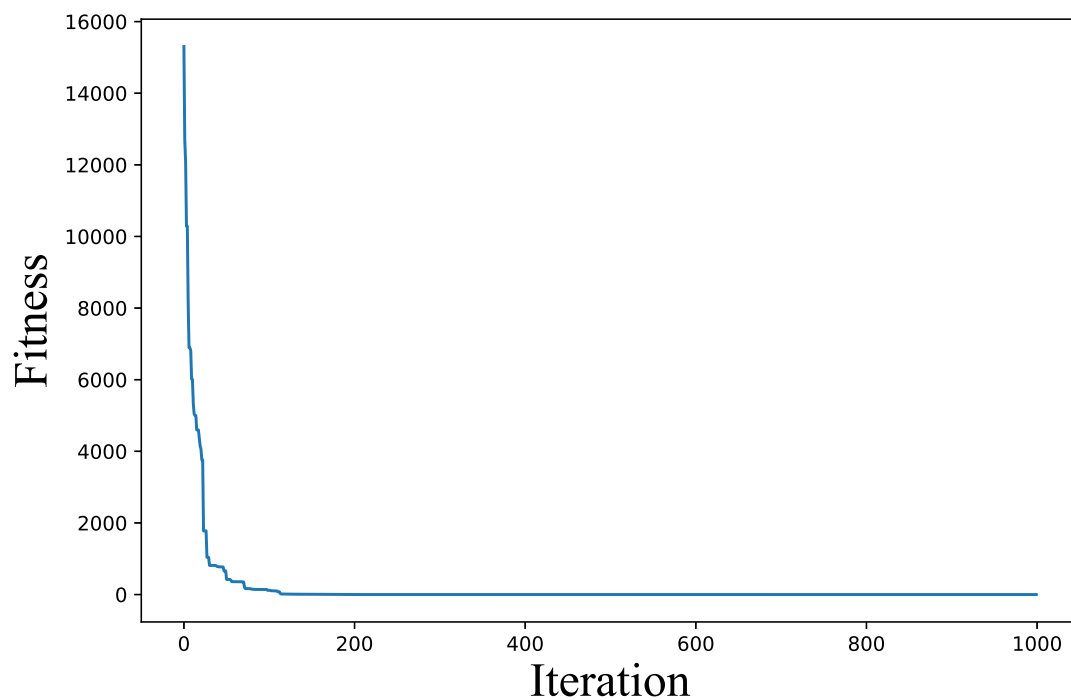


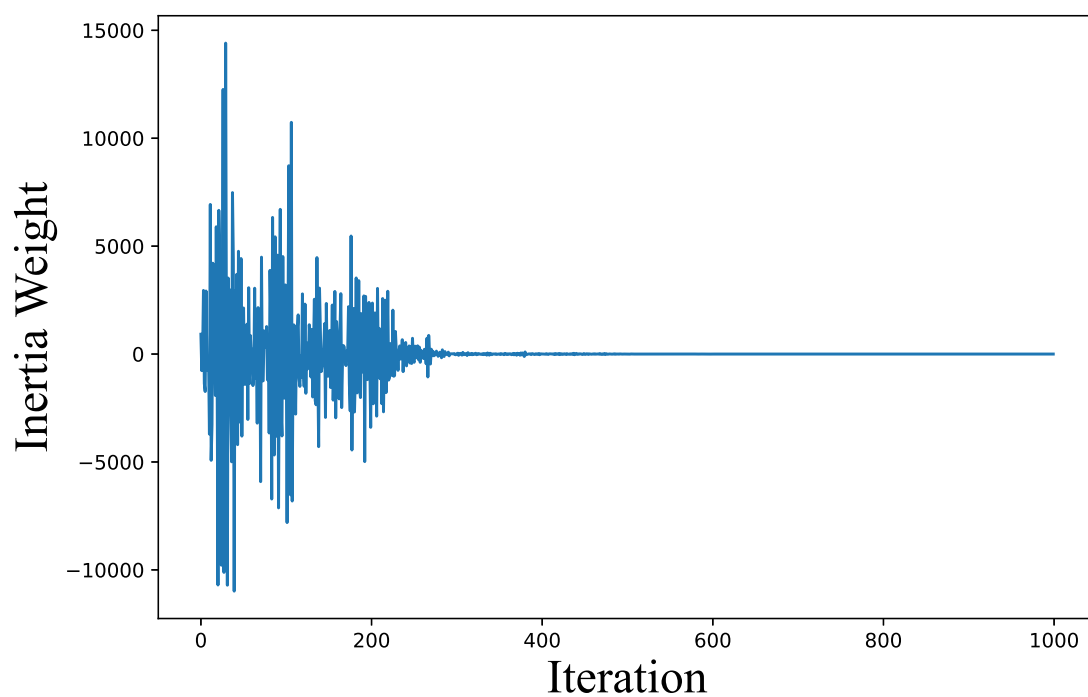
Table 1: Parameter of ACO

Parameter	Value
Number of Particles	20
$C_1$	2
$C_2$	2
$w$	0.5

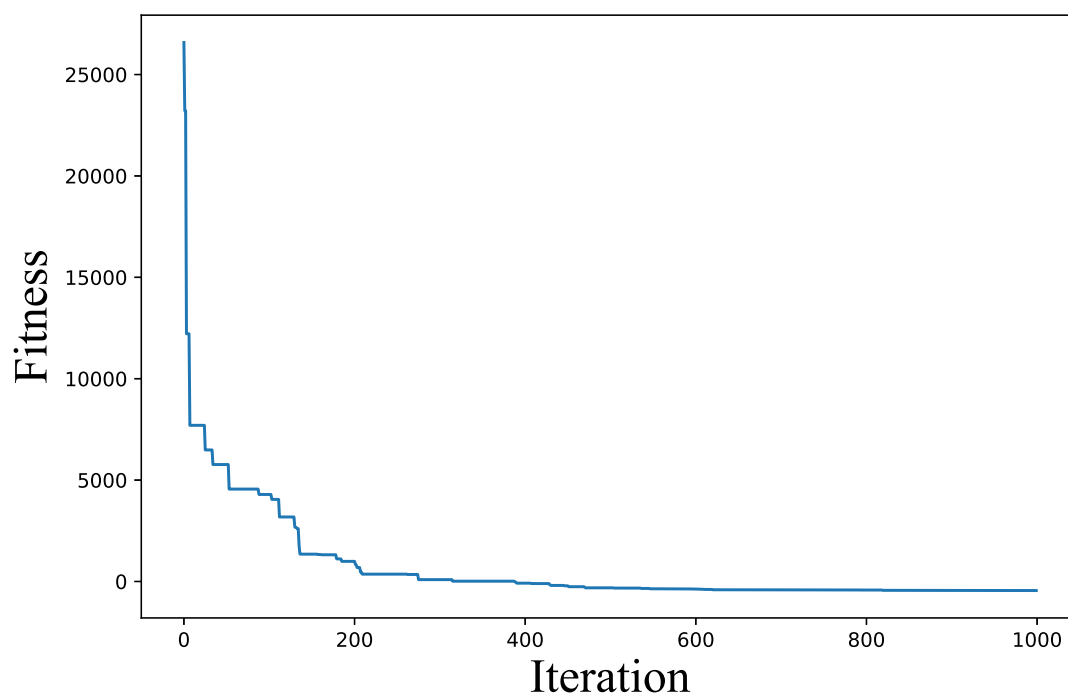
شکل ۲: نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 10$ ) برای ۱۰۰۰ تکرار



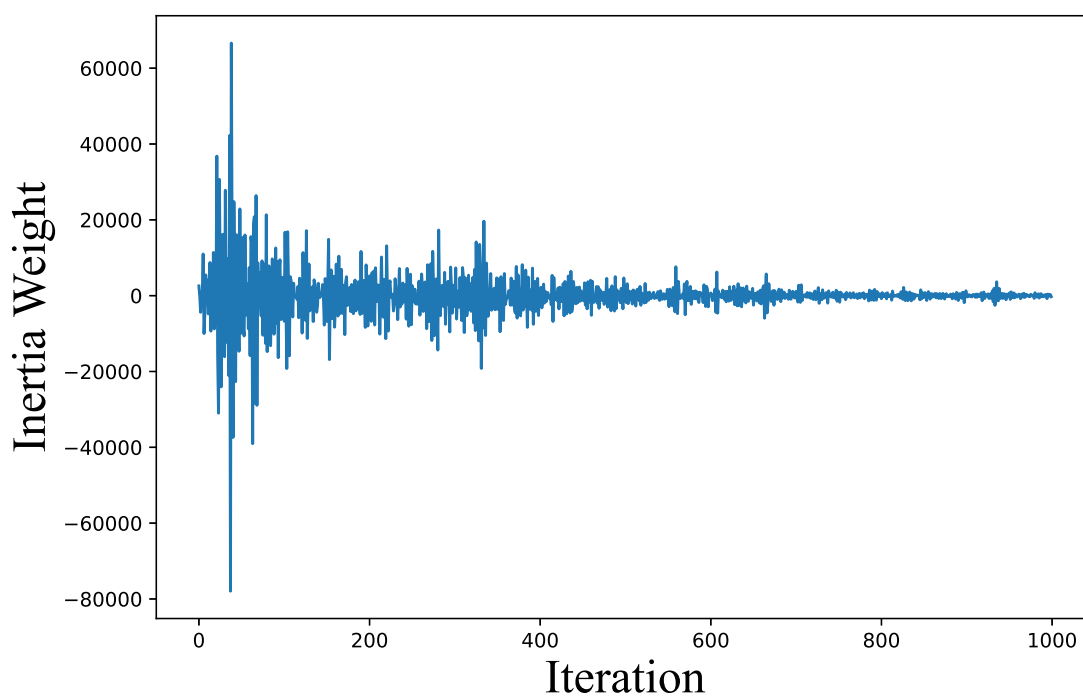
شکل ۳: نمودار inertia weight الگوریتم PSO تابع شماره یک ( $D = 10$ ) برای ۱۰۰۰ تکرار



شکل ۴: نمودار همگرایی الگوریتم PSO تابع شماره دو ( $D = 10$ ) برای ۱۰۰۰ تکرار



شکل ۵: نمودار inertia weight الگوریتم PSO تابع شماره دو ( $D = 10$ ) برای ۱۰۰۰ تکرار



به علت اینکه تابع شماره دو دارای نویز است، پس، inertia weight آن نیز دارای نویز است.

Table 2: Values Achieved with PSO algorithm for Problems 1 and 2 (D=10)

FES/Problem		Problem 1	Problem 2
1e3	1 <sup>th</sup> (Best)	-449.9999787052277	-449.99999999999955
	7 <sup>th</sup>	-449.999853416358	-449.9999999983805
	13 <sup>th</sup> (Median)	-449.9996680779018	-449.999985766405
	19 <sup>th</sup>	-449.9992664160519	-449.8254439788584
	25 <sup>th</sup> (Worst)	-449.9935680412684	-61.45240223709476
	Mean	-449.9992093850963	-423.450611298781
	Std	0.001321429062858954	81.2593245611484
1e4	1 <sup>th</sup> (Best)	-450.0	-449.99999999999966
	7 <sup>th</sup>	-450.0	-449.999999990536
	13 <sup>th</sup> (Median)	-449.9999999999994	-449.9890024477813
	19 <sup>th</sup>	-449.9999999999994	-429.92744443669807
	25 <sup>th</sup> (Worst)	-449.9999999999983	4782.609097090884
	Mean	-450.0	-129.8996247758574
	Std	5.796914039811765e-14	1068.5144300632844
1e5	1 <sup>th</sup> (Best)	-450.0	-450.0
	7 <sup>th</sup>	-450.0	-449.9999999998863
	13 <sup>th</sup> (Median)	-450.0	-449.9997990210992
	19 <sup>th</sup>	-449.9999999999994	-449.923712929338
	25 <sup>th</sup> (Worst)	-449.9999999999994	473.6923826368156
	Mean	-450.0	-409.6836050765505
	Std	3.215549355384371e-14	180.6429570540459

Table 3: Values Achieved with PSO algorithm for Problems 1 and 2 (D=30)

FES/Problem		Problem 1	Problem 2
1e3	1 <sup>th</sup> (Best)	-449.99998619166854	-449.99999999939354
	7 <sup>th</sup>	-449.9998718525022	40420.10955750148
	13 <sup>th</sup> (Median)	-449.9998195592621	56691.59162357271
	19 <sup>th</sup>	-449.99942177358423	79561.30101108812
	25 <sup>th</sup> (Worst)	-449.9985720932928	123160.65222491047
	Mean	-449.99962402843727	58183.77085117807
	Std	0.0003711894716529072	30723.13488958691
1e4	1 <sup>th</sup> (Best)	-450.0	23595.40655734655
	7 <sup>th</sup>	-450.0	33363.7352014709
	13 <sup>th</sup> (Median)	-449.99999999999994	47183.89788197362
	19 <sup>th</sup>	-449.99999999999994	60886.23378363138
	25 <sup>th</sup> (Worst)	-449.99999999999994	103569.85779628623
	Mean	-450.0	49939.64569103153
	Std	4.5474735088646414e-14	21155.063674898593
1e5	1 <sup>th</sup> (Best)	-450.0	42625.903518764586
	7 <sup>th</sup>	-450.0	76377.64509731466
	13 <sup>th</sup> (Median)	-450.0	98728.87253737042
	19 <sup>th</sup>	-449.99999999999994	128860.87687077752
	25 <sup>th</sup> (Worst)	-449.99999999999994	179366.99419346155
	Mean	-450.0	99025.05071090581
	Std	3.215549355384371e-14	34041.49636204158

Table 4: Values Achieved with PSO algorithm for Problems 1 and 2 (D=50)

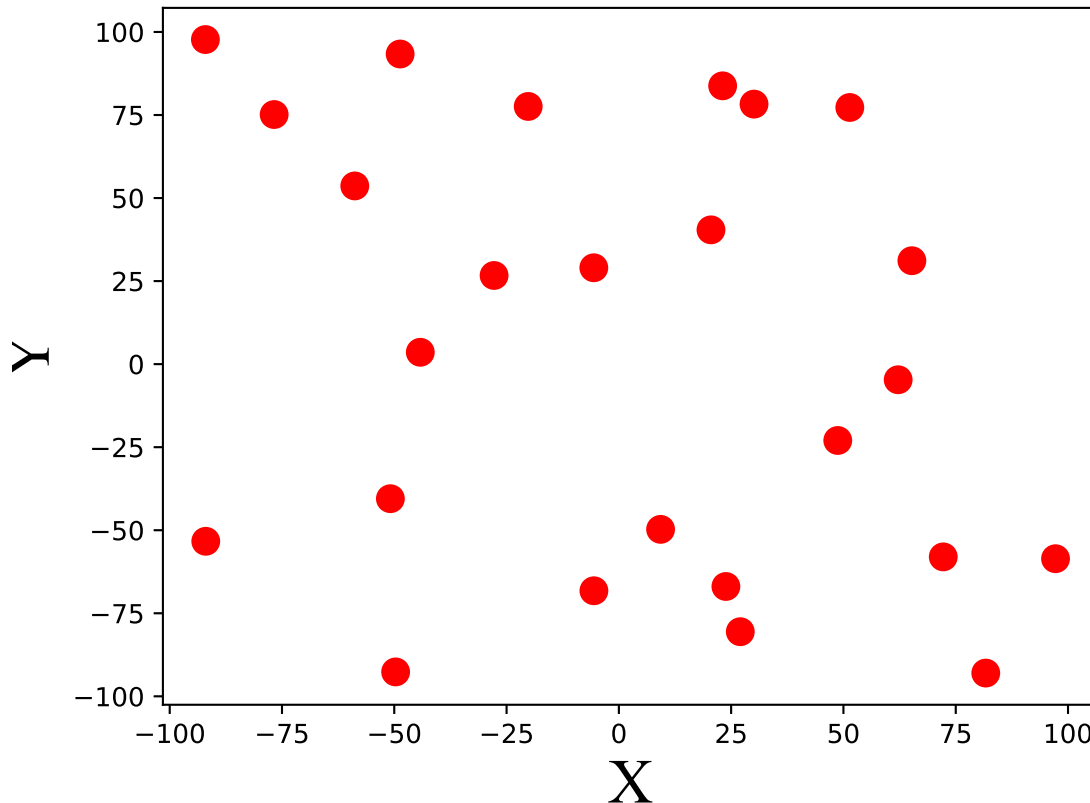
FES/Problem		Problem 1	Problem 2
1e3	1 <sup>th</sup> (Best)	-449.99998100894874	18439.04388319432
	7 <sup>th</sup>	-449.9999238343942	44613.926387444895
	13 <sup>th</sup> (Median)	-449.9998536654368	69394.46049754831
	19 <sup>th</sup>	-449.9996903867559	96414.65498879585
	25 <sup>th</sup> (Worst)	-449.99863762582254	153348.96108922383
	Mean	-449.99973406754907	69754.01098699088
	Std	0.0003002225766163943	37537.44745190397
1e4	1 <sup>th</sup> (Best)	-450.0	123434.7542796172
	7 <sup>th</sup>	-450.0	161688.4767345813
	13 <sup>th</sup> (Median)	-449.99999999999994	196405.68546559024
	19 <sup>th</sup>	-449.99999999999994	260789.9331202219
	25 <sup>th</sup> (Worst)	-449.99999999999994	380963.02282124246
	Mean	-450.0	215538.32784349122
	Std	4.5474735088646414e-14	73059.88366852782
1e5	1 <sup>th</sup> (Best)	-449.99999999999994	153218.21508463015
	7 <sup>th</sup>	-449.99999999999983	205351.82441262604
	13 <sup>th</sup> (Median)	-449.99999999999994	319569.8014374089
	19 <sup>th</sup>	-449.99999999999935	426984.8566112531
	25 <sup>th</sup> (Worst)	-449.9998364102162	595897.6275575941
	Mean	-449.9999932417356	343549.239596291
	Std	3.202581535459253e-05	135620.64779918295

## ۲ سوال دوم

در این سوال از الگوریتم ابتکاری مورچگان برای حل مسئله معروف فروشنده دوره‌گرد استفاده شده است. بخش ۱.۲ با فرض نبودن ترافیک، بخش ۲.۲ با فرض ترافیک و بخش ۳.۲ با فرض اینکه

ترافیک تابعی از زمان است حل شده است. در شکل ۶ مکان شهرها رسم شده است.

شکل ۶: مکان شهرها



در الگوریتم مورچگان ابتدا ماتریس فرمون برای هر مسیر تعریف می‌شود. با توجه به اینکه هیچ مورچه‌ای هنوز حرکت نکرده است، در ماتریس فرمون تعریف شده هر مسیر فرمون برابر با یک است. در ادامه هر مورچه برای انتخاب مسیر از احتمال زیر استفاده می‌کند.

$$p_n(c_i, c_j) = \frac{\tau(c_i, c_j)^\alpha / \delta(c_i, c_j)^\beta}{\sum \tau(c_i, c_j)^\alpha / \delta(c_i, c_j)^\beta} \quad (3)$$

که در رابطه بالا  $\tau$  برابر با مقدار فرمون هر مسیر و  $\delta$  برابر با طول مسیر است. بعد از آنکه تمامی مورچه‌ها حرکت کردند فرمون مسیرها به صورت زیر به‌روز رسانی می‌شود.

$$\tau(c_i, c_j) = (1 - \rho)\tau(c_i, c_j) + \sum_{n=1}^m \Delta\tau(c_i, c_j) \quad (4)$$

که پارامتر رابطه بالا به صورت زیر تعریف می‌شود.

$$\Delta\tau(c_i, c_j) = \frac{Q}{L} \quad (5)$$



که در رابطه بالا  $L$  برابر با طول مسیر طی شده توسط مورچه است. پارامتر  $Q$  برای رفتار بهتر الگوریتم استفاده شده است. پارامترهای الگوریتم مورچگان بر اساس اعداد رایج در مقاله‌ها انتخاب شده است. پارامترهای الگوریتم مورچگان در جدول ۵ آورده شده است.

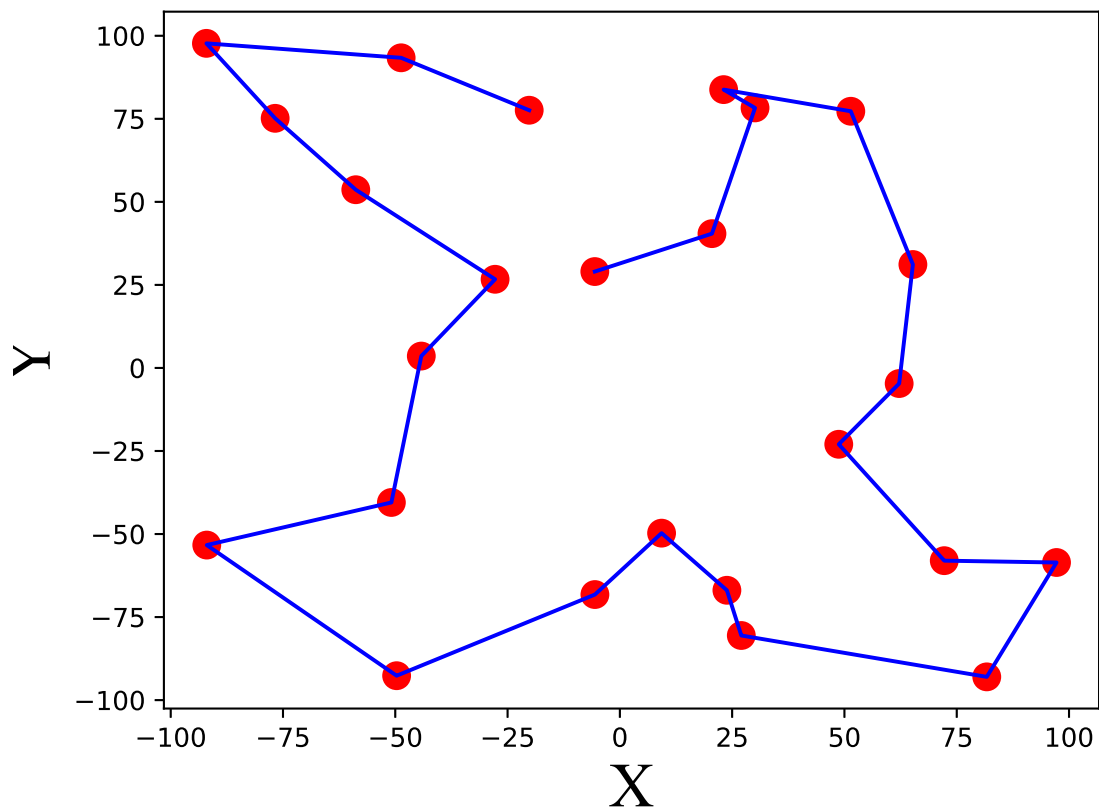
Table 5: Parameter of ACO

Parameter	Value
Number of Ants	10
Number of Iterations	100
$\alpha$	1
$\beta$	5
$\rho$	0.5
$Q$	100

## ۱.۲ پخش اول

با توجه به اینکه الگوریتم مورد استفاده ابتکاری است و تابع رندوم بخش مهمی از آن است در اینجا دو راه حل (شکل‌های ۱.۲ و ۱.۲) آورده شده است. طول هر مسیر نیز در جدول ۱.۲ آورده شده است. برای حل این مسئله از یک ماتریس به اسم graph برای توصیف فاصله‌ی بین شهرها استفاده شده است

شکل ۷: راه حل اول تولید شده توسط الگوریتم ACO



شکل ۸: راه حل دوم تولید شده توسط الگوریتم ACO

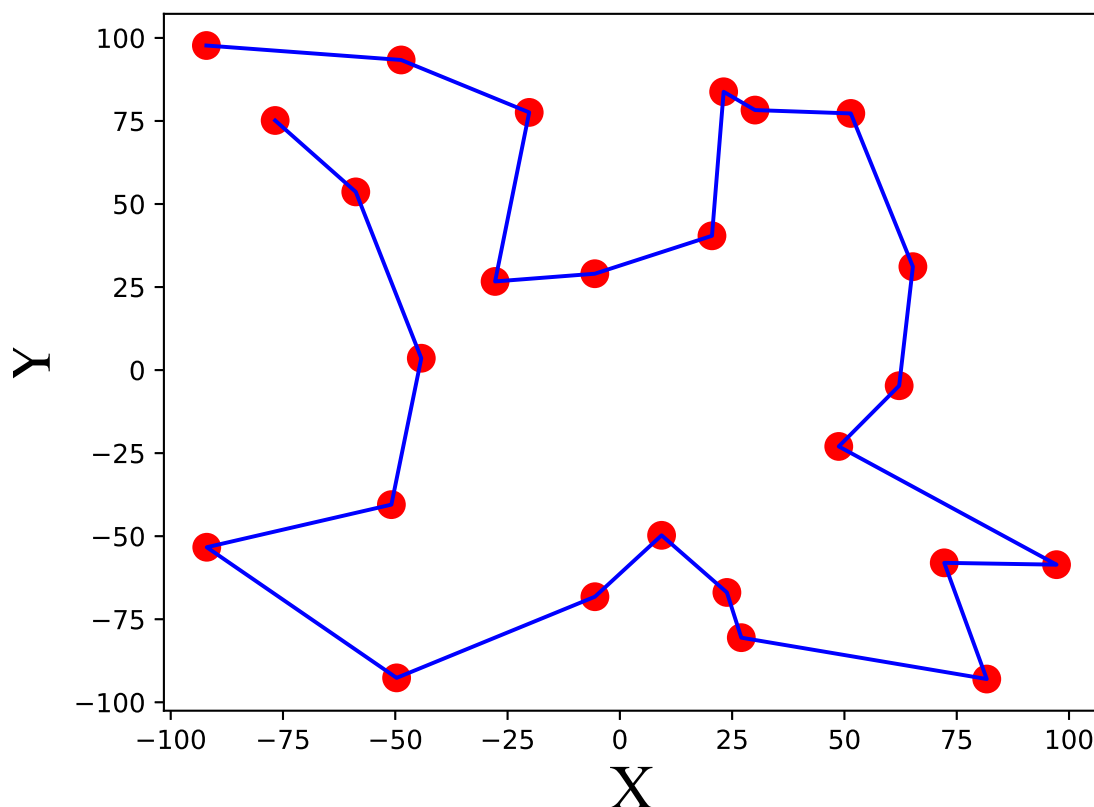


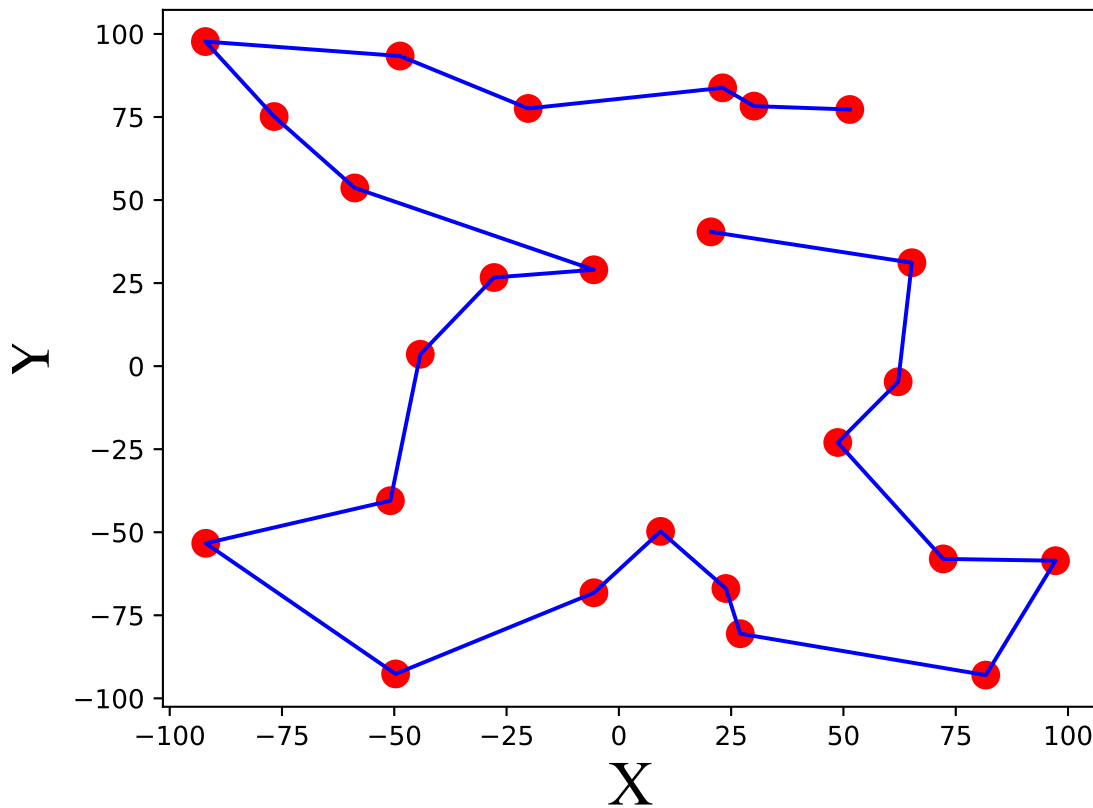
Table 6: Cost of solution produced with ACO

solution 1 Cost	solution 2 Cost
881.0	899.8

## ۲.۲ بخش دوم

در بخش ۱.۲ ماتریس graph معرفی شد. در این بخش برای اضافه کردن ترافیک، درایه‌های ماتریس traffic به صورت نظیر به نظیر تقسیم بر درایه‌های ماتریس graph (هر چه سرعت بیشتر باشد هزینه رفتن از شهر i به j کمتر است) می‌شوند.

شکل ۹: راه حل تولید شده توسط الگوریتم ACO با در نظر گرفتن ترافیک



## ۳.۲ بخش سوم

برای ترافیک متغیر با زمان دو رویکرد در نظر گرفته شد. رویکرد اول آن بود که در هر زمان مسئله از اول حل شود و رویکرد دوم به این صورت بود که در زمان بعدی فرمون زمان قبل باقی بماند ولی ترافیک مسیر عوض شده باشد. مقایسه نتایج دو رویکرد در جدول ۳.۲ آورده شده است. برای پویانمایی مسیر بهینه دو فایل gif با نام‌های `time_varing_traffic_solution` و `time_varing_traffic_solution_gph` در پوشه `Figure/Q2` وجود دارد که هر کدام برای یک رویکرد اشاره شده است.

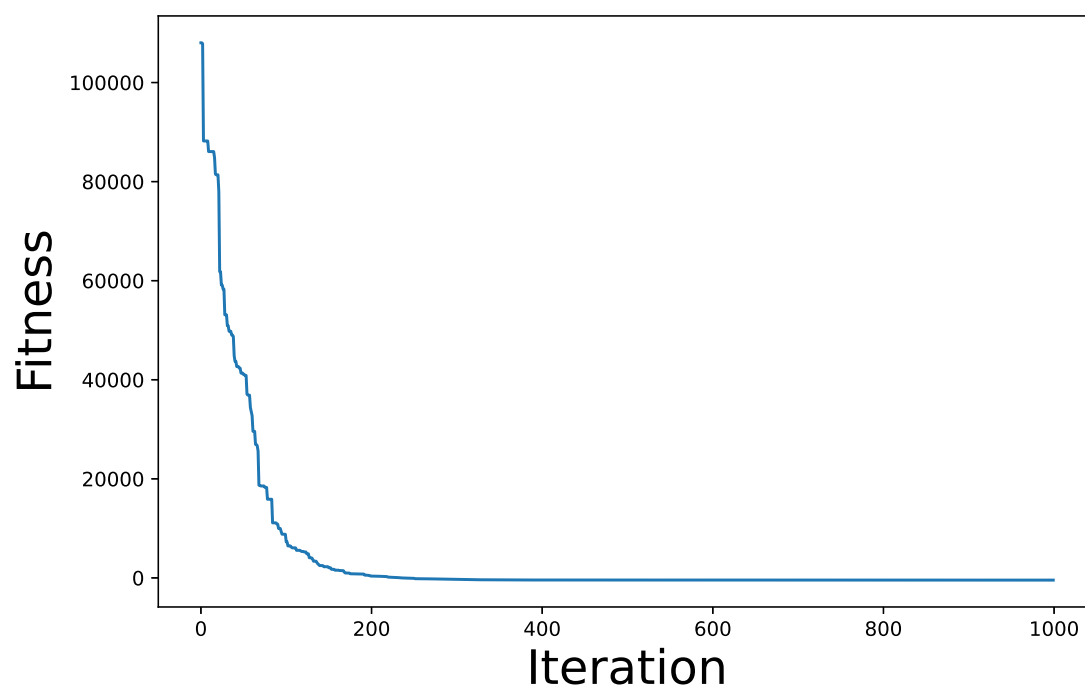
Table 7: Cost of solution produced with ACO

Time	local pheromone solution	global pheromone solution
i = 0	874.72	879.96
i = 1	1491.50	1571.60
i = 2	936.80	1138.58
i = 3	1130.20	1152.92
i = 4	1011.32	1095.87
i = 5	1030.46	1080.46
i = 6	1324.29	1357.68
i = 7	1081.62	1077.28
i = 8	1129.28	1163.53
i = 9	1329.87	1309.49
sum	11340.11	11827.44

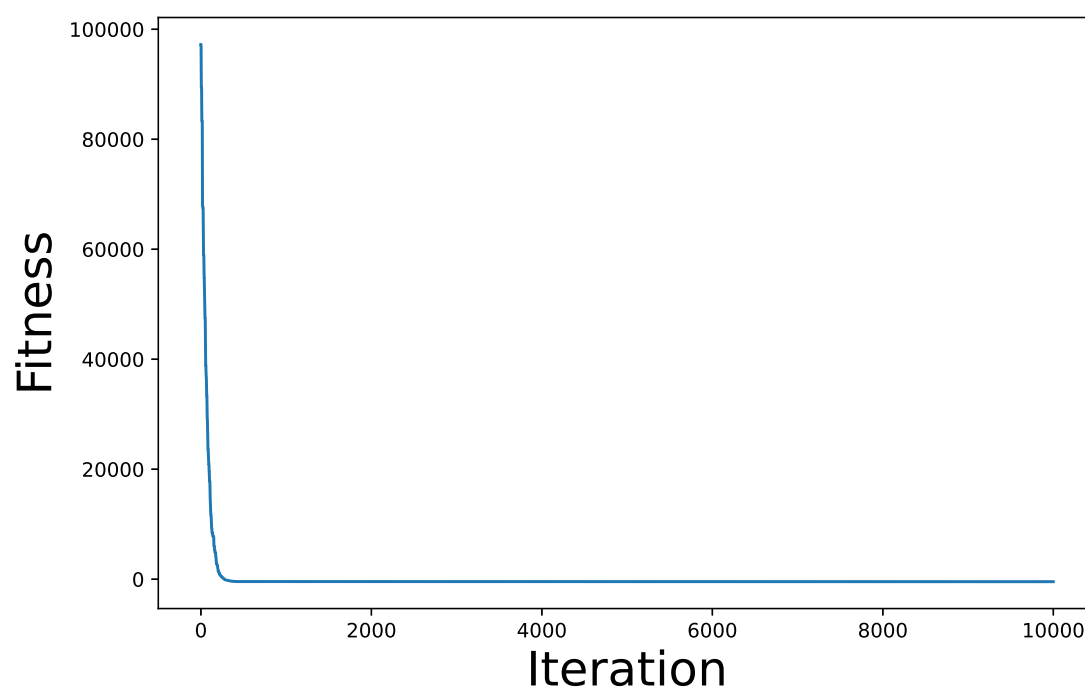
## ۳ پیوست

در این بخش نمودارهای الگوریتم PSO آورده شده است.

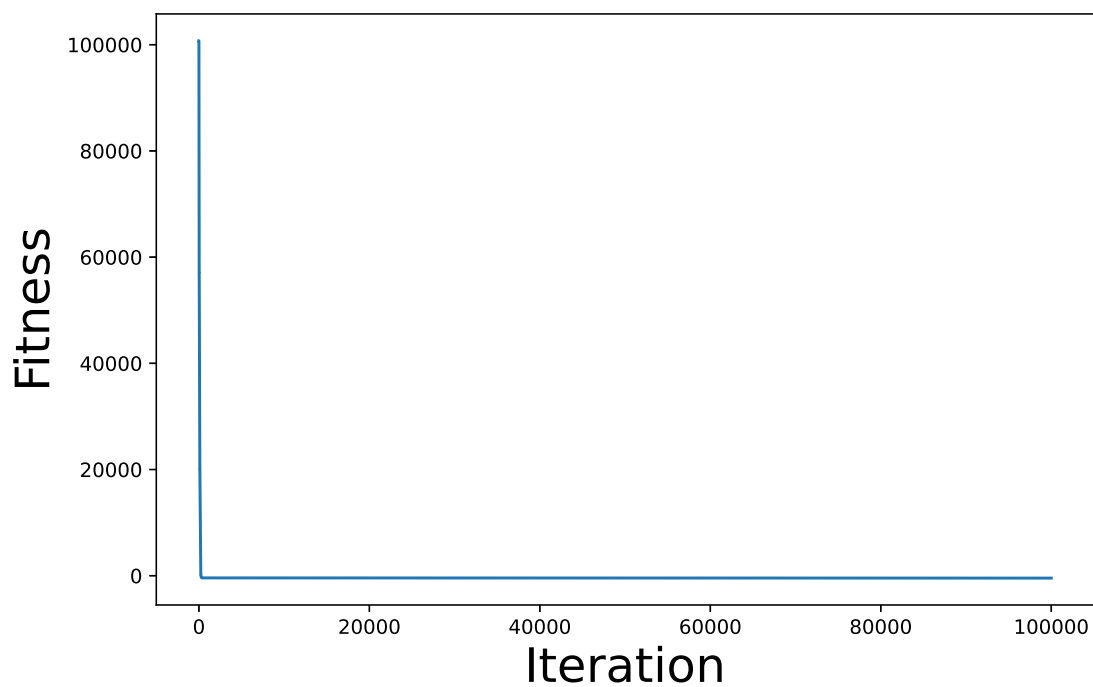
شکل ۱۰: نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 10$ ) برای ۱۰۰۰ تکرار



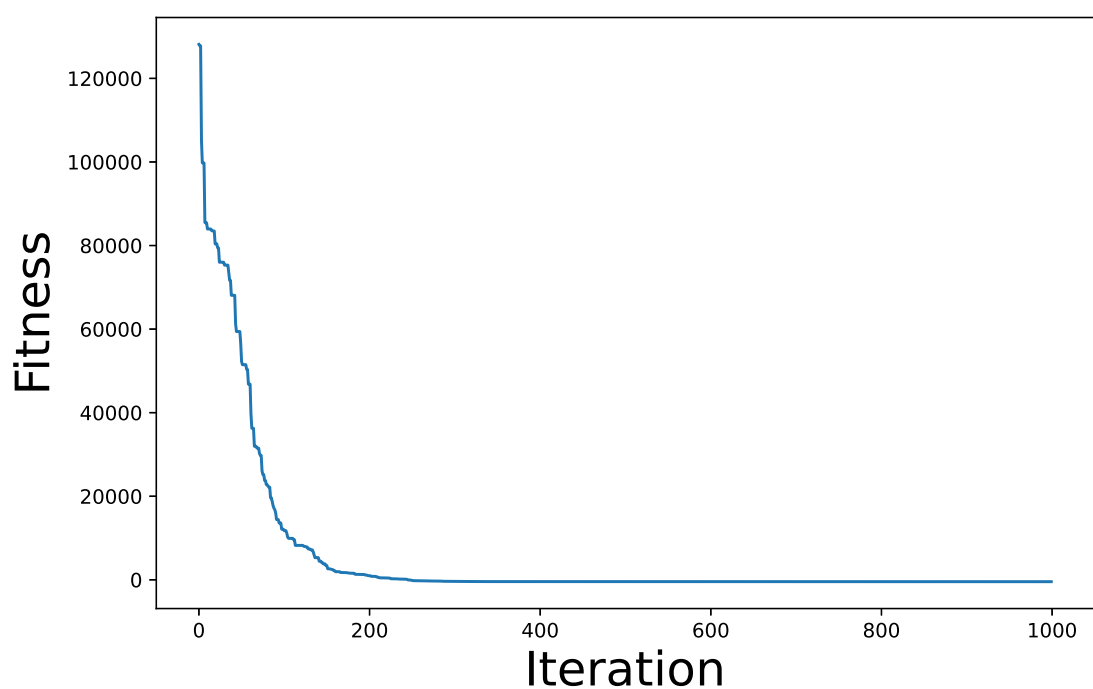
شکل ۱۱: نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 10$ ) برای ۱۰۰۰۰ تکرار



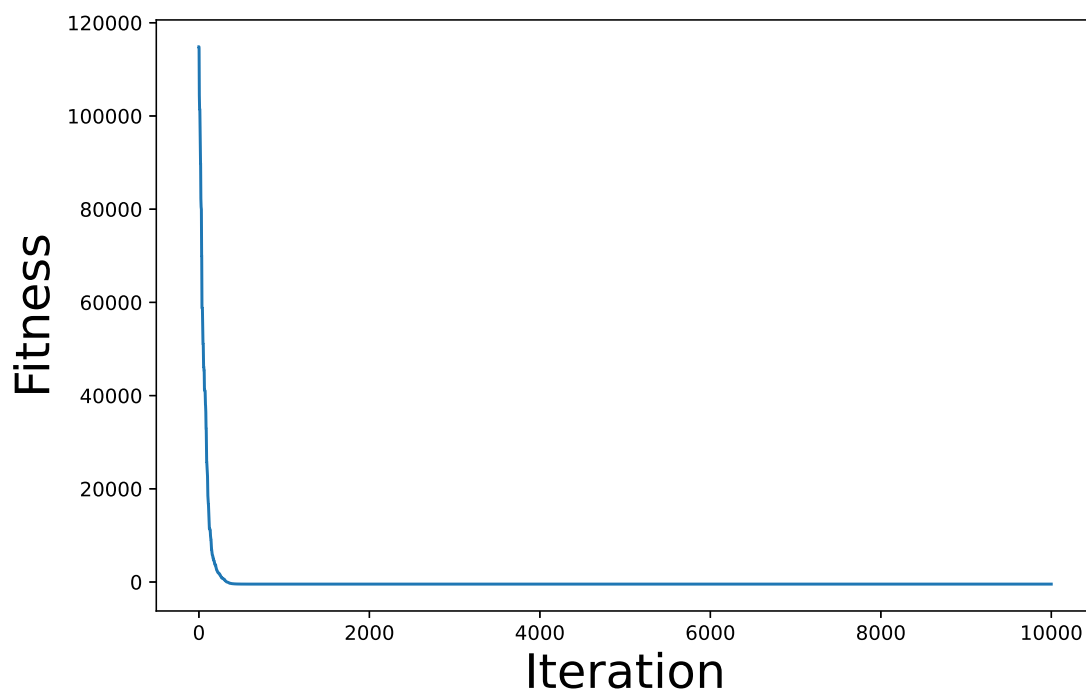
شکل ۱۲: نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 10$ ) برای ۱۰۰۰۰۰ تکرار



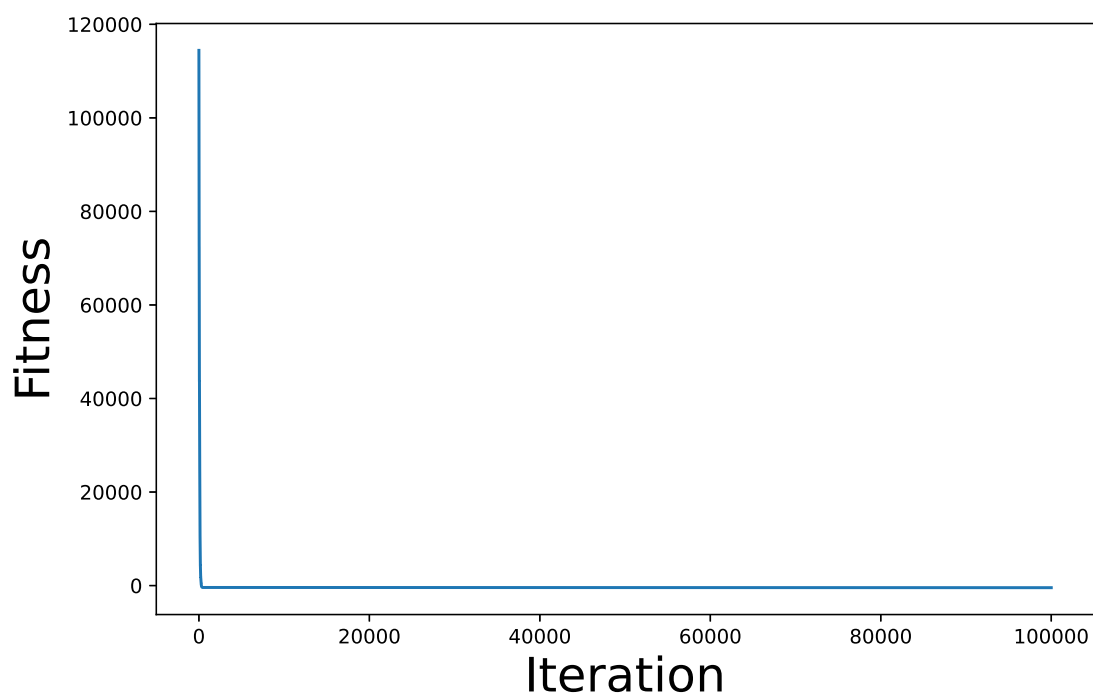
شکل ۱۳: نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 30$ ) برای ۱۰۰۰ تکرار



شکل ۱۴: نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 30$ ) برای ۱۰۰۰۰ تکرار

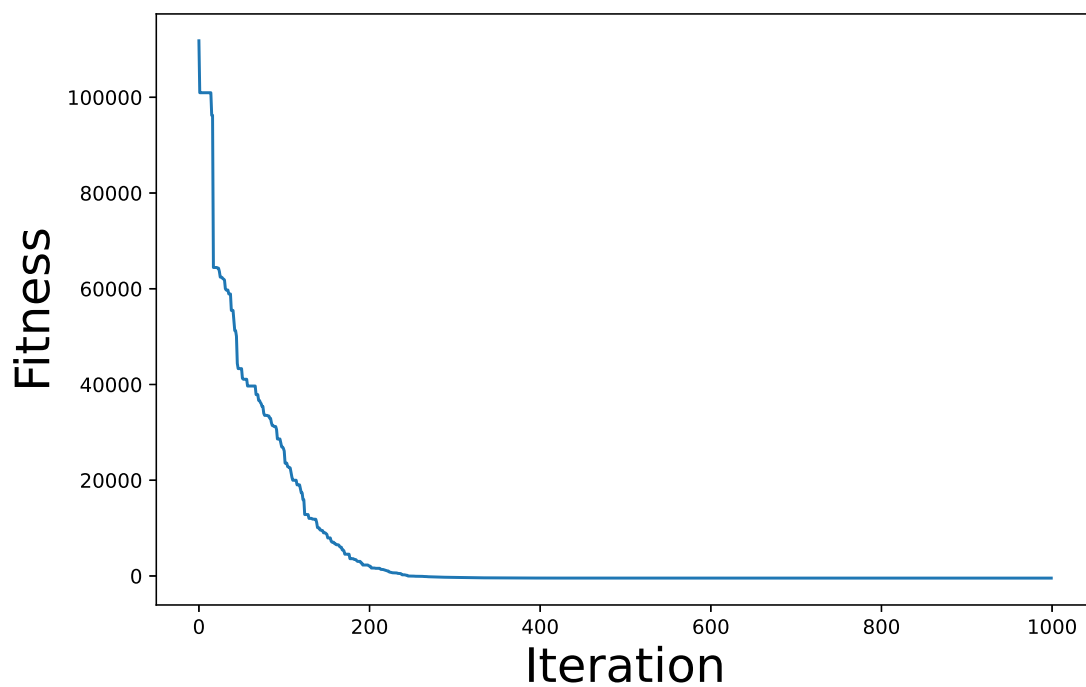


شکل ۱۵: نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 30$ ) برای ۱۰۰۰۰۰ تکرار

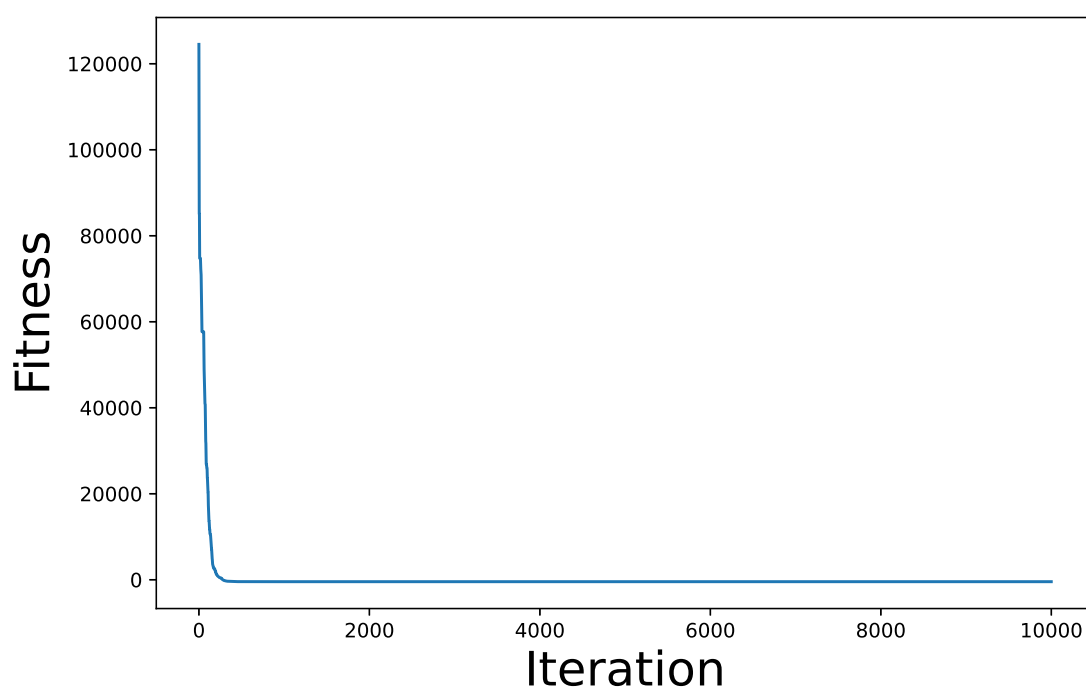




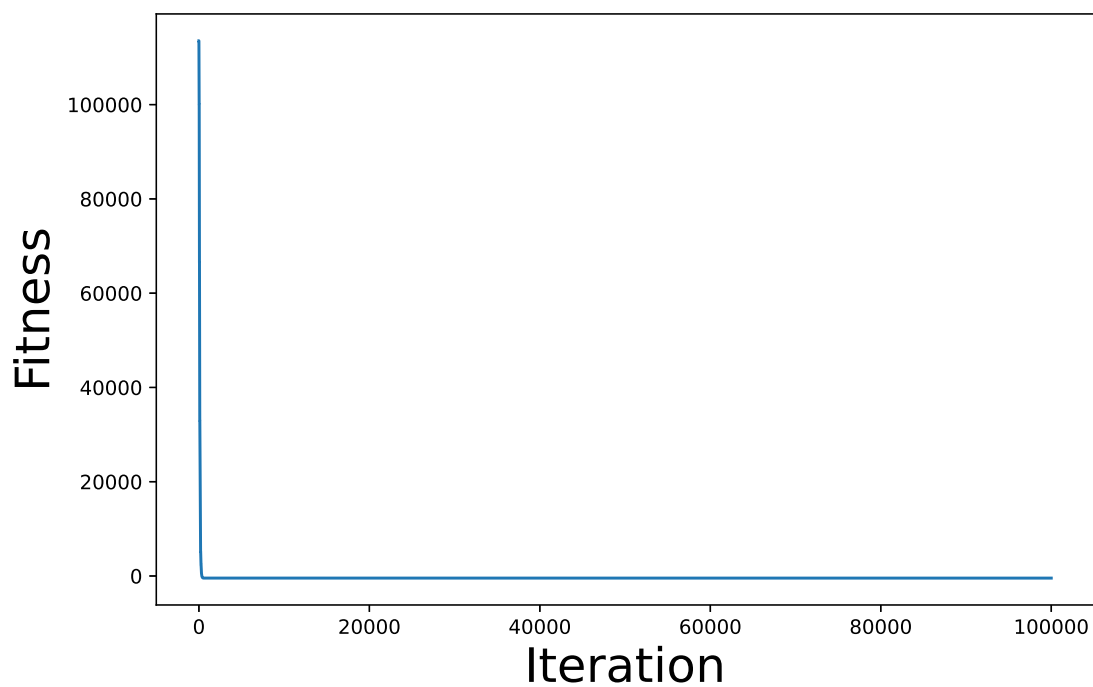
شکل ۱۶: نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 50$ ) برای ۱۰۰۰ تکرار



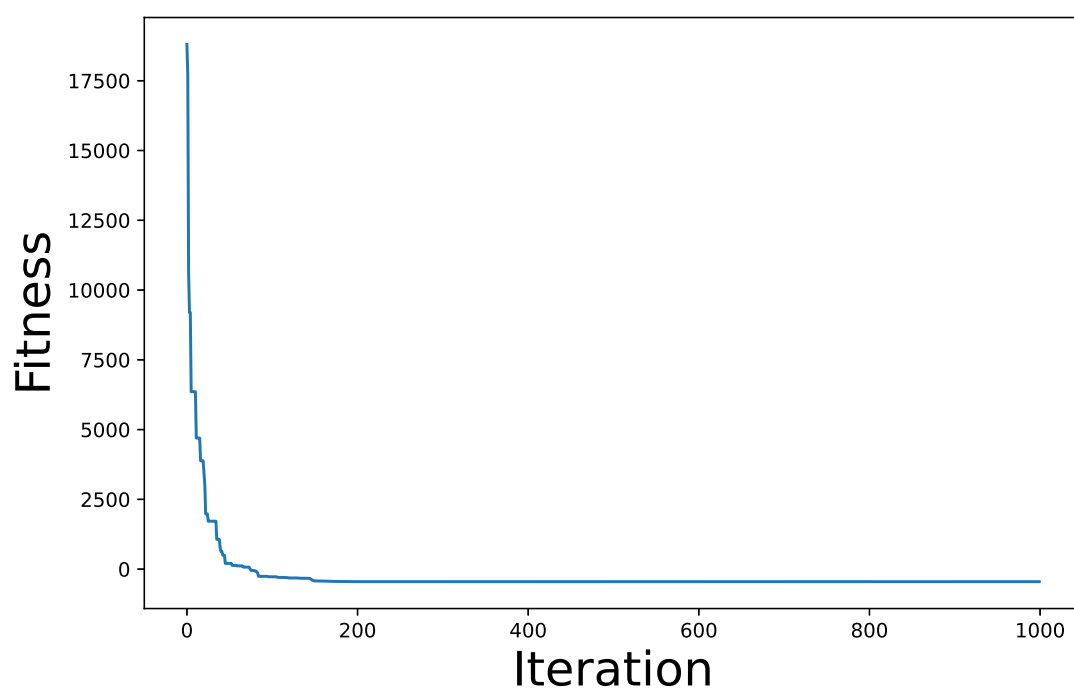
شکل ۱۷: نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 50$ ) برای ۱۰,۰۰۰ تکرار



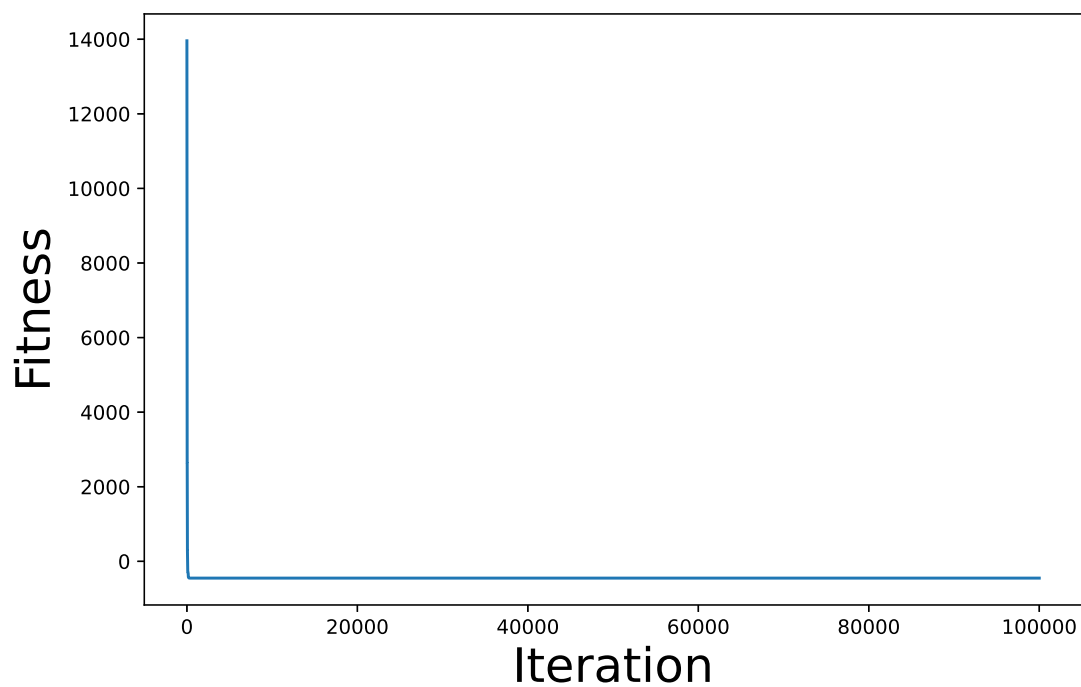
شکل ۱۸: نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 50$ ) برای ۱۰۰۰۰۰ تکرار



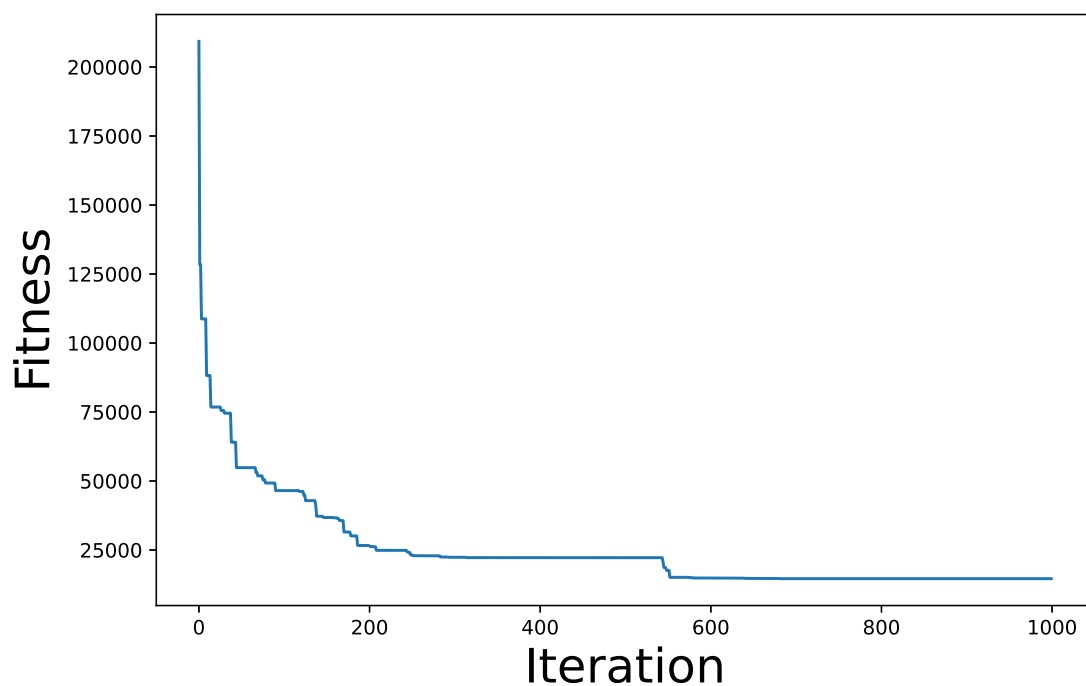
شکل ۱۹: نمودار همگرایی الگوریتم PSO تابع شماره دو ( $D = 10$ ) برای ۱۰۰۰ تکرار



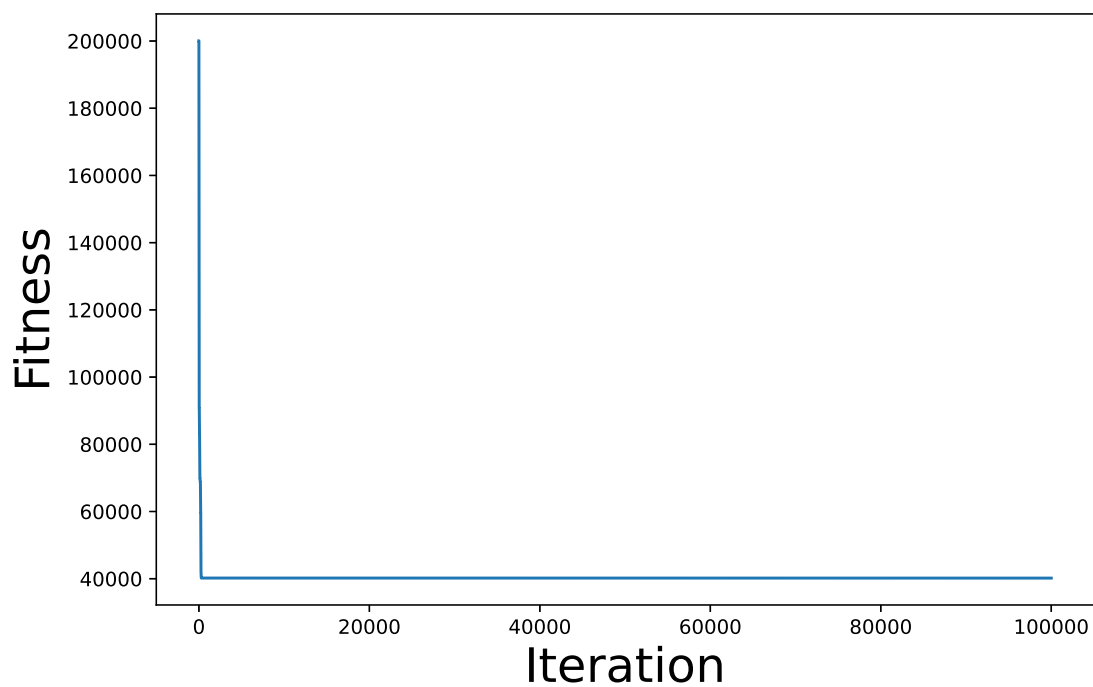
شکل ۲۰: نمودار همگرایی الگوریتم PSO تابع شماره دو ( $D = 10$ ) برای ۱۰۰۰۰۰ تکرار



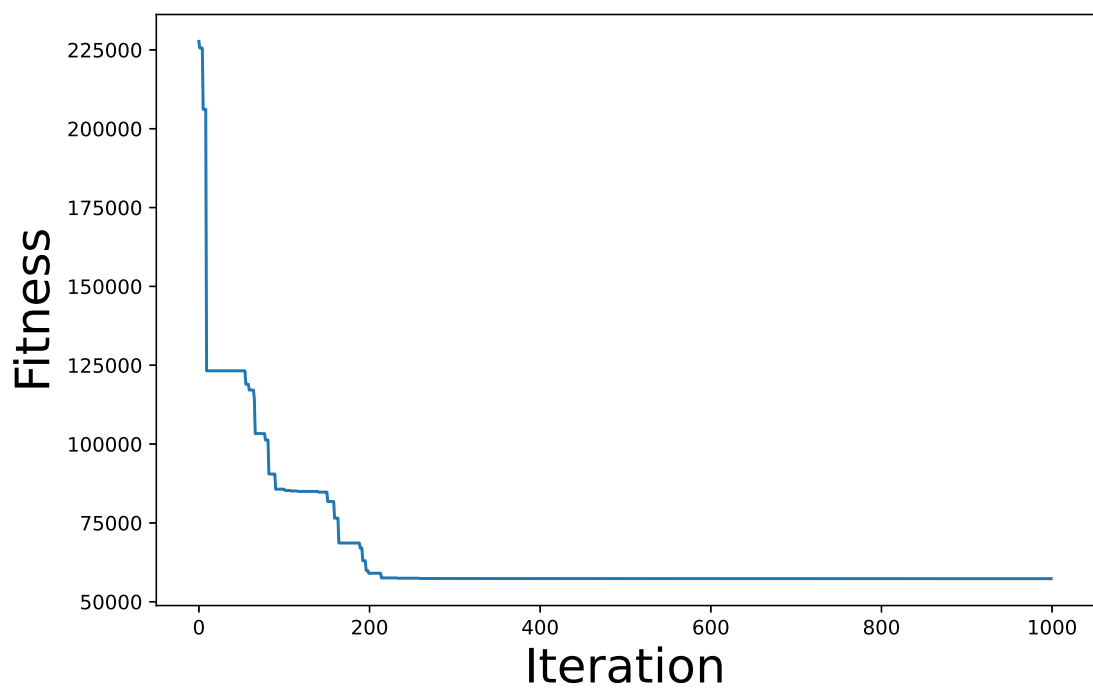
شکل ۲۱: نمودار همگرایی الگوریتم PSO تابع شماره دو ( $D = 30$ ) برای ۱۰۰۰ تکرار



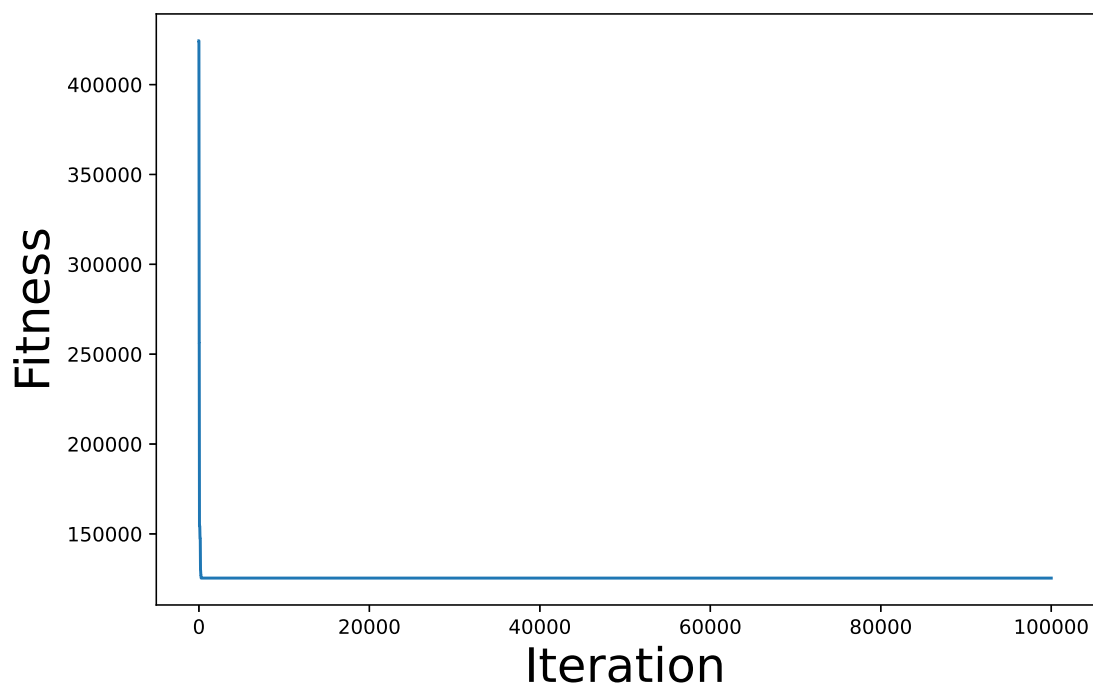
شکل ۲۲: نمودار همگرایی الگوریتم PSO تابع شماره دو ( $D = 30$ ) برای ۱۰۰۰۰۰ تکرار



شکل ۲۳: نمودار همگرایی الگوریتم PSO تابع شماره دو ( $D = 50$ ) برای ۱۰۰۰ تکرار



شکل ۲۴: نمودار همگرایی الگوریتم PSO تابع شماره دو ( $D = 50$ ) برای ۱۰۰۰۰۰ تکرار



## فهرست مطالب

۱	سوال اول	۱
۷	سوال دوم	۲
۹	۱.۲ بخش اول	۱۰.۲
۱۱	۲.۲ بخش دوم	۲۰.۲
۱۲	۳.۲ بخش سوم	۳۰.۲
۱۳	پیوست	۳

## فهرست تصاویر

۱	توپولوژی همسایگی حلقه	۲
۲	نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 10$ ) برای ۱۰۰۰ تکرار	۳
۳	نمودار inertia weight الگوریتم PSO تابع شماره یک ( $D = 10$ ) برای ۱۰۰۰ تکرار	۳
۴	نمودار همگرایی الگوریتم PSO تابع شماره دو ( $D = 10$ ) برای ۱۰۰۰ تکرار	۴
۵	نمودار inertia weight الگوریتم PSO تابع شماره دو ( $D = 10$ ) برای ۱۰۰۰ تکرار	۴
۶	مکان شهرها	۸
۷	راه حل اول تولید شده توسط الگوریتم ACO	۱۰
۸	راه حل دوم تولید شده توسط الگوریتم ACO	۱۱
۹	راه حل تولید شده توسط الگوریتم ACO با در نظر گرفتن ترافیک	۱۲
۱۰	نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 10$ ) برای ۱۰۰۰ تکرار	۱۴
۱۱	نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 10$ ) برای ۱۰۰۰۰ تکرار	۱۴
۱۲	نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 10$ ) برای ۱۰۰۰۰۰ تکرار	۱۵
۱۳	نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 30$ ) برای ۱۰۰۰ تکرار	۱۵
۱۴	نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 30$ ) برای ۱۰۰۰۰ تکرار	۱۶
۱۵	نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 30$ ) برای ۱۰۰۰۰۰ تکرار	۱۶
۱۶	نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 50$ ) برای ۱۰۰۰ تکرار	۱۷
۱۷	نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 50$ ) برای ۱۰۰۰۰ تکرار	۱۷
۱۸	نمودار همگرایی الگوریتم PSO تابع شماره یک ( $D = 50$ ) برای ۱۰۰۰۰۰ تکرار	۱۸
۱۹	نمودار همگرایی الگوریتم PSO تابع شماره دو ( $D = 10$ ) برای ۱۰۰۰ تکرار	۱۸
۲۰	نمودار همگرایی الگوریتم PSO تابع شماره دو ( $D = 10$ ) برای ۱۰۰۰۰۰ تکرار	۱۹
۲۱	نمودار همگرایی الگوریتم PSO تابع شماره دو ( $D = 30$ ) برای ۱۰۰۰ تکرار	۱۹
۲۲	نمودار همگرایی الگوریتم PSO تابع شماره دو ( $D = 30$ ) برای ۱۰۰۰۰۰ تکرار	۲۰
۲۳	نمودار همگرایی الگوریتم PSO تابع شماره دو ( $D = 50$ ) برای ۱۰۰۰ تکرار	۲۰
۲۴	نمودار همگرایی الگوریتم PSO تابع شماره دو ( $D = 50$ ) برای ۱۰۰۰۰۰ تکرار	۲۱

## List of Tables

1	Parameter of ACO . . . . .	2
2	Values Achieved with PSO algorithm for Problems 1 and 2 (D=10) . . .	5
3	Values Achieved with PSO algorithm for Problems 1 and 2 (D=30) . . .	6
4	Values Achieved with PSO algorithm for Problems 1 and 2 (D=50) . . .	7
5	Parameter of ACO . . . . .	9
6	Cost of solution produced with ACO . . . . .	11
7	Cost of solution produced with ACO . . . . .	13