EVOLUTIONARY ALGORITHMS PROJECT 3

Function Rosenbrock:

a=1,5 b=1,5

$$f(x) = (1 - x - a)^2 + 100 * (y - b - (x - a)^2)^2$$

Min(x,y)=(-0.5,5.5)

f(-0.5,5.5)=0

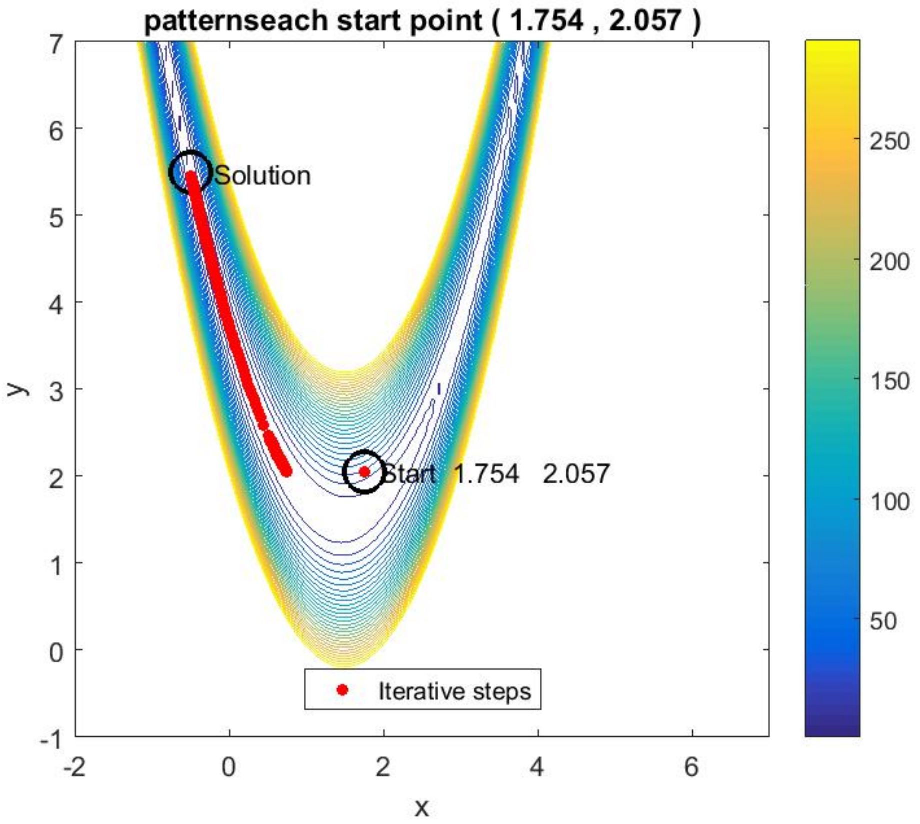
Summarize all results:

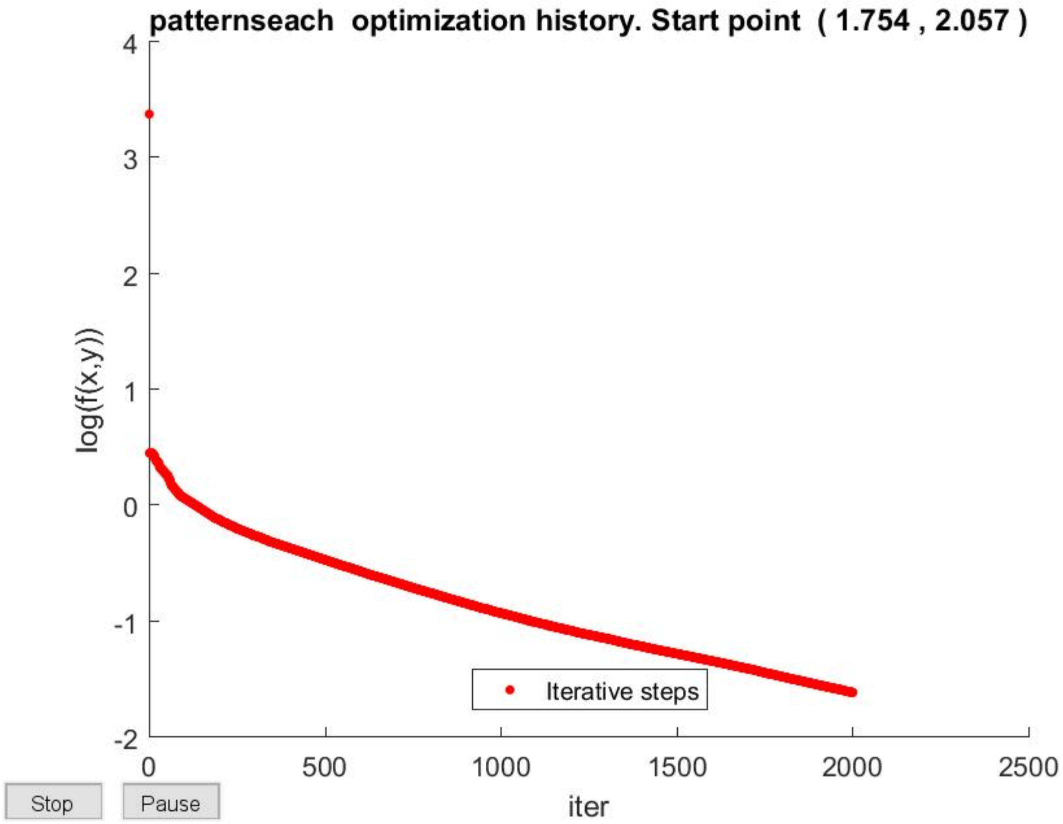
<table>
<tr><td>Algorithm</td><td>x</td><td>y</td><td>Nº f evaluations</td><td>Nº solver iterations</td><td colspan="2">F min</td></tr>
<tr><td rowspan="4">PatterSearch</td><td>1,753</td><td>2,056</td><td>88298</td><td>27860</td><td colspan="2">7,349e-06</td></tr>
<tr><td>3,326</td><td>2,593</td><td>114440</td><td>36264</td><td colspan="2">3,586e-05</td></tr>
<tr><td>2,764</td><td>3,415</td><td>49696</td><td>15262</td><td colspan="2">3,057e-06</td></tr>
<tr><td>1,695</td><td>3,429</td><td>48461</td><td>14898</td><td colspan="2">2,027e-06</td></tr>
<tr><td rowspan="8">GA</td><td colspan="2">Parameters</td><td>PopulationSize</td><td>Nº solver iterations</td><td>Total</td><td>F min</td></tr>
<tr><td colspan="2">1ºregulation</td><td>50</td><td>83</td><td>4015</td><td>1,00e-06</td></tr>
<tr><td colspan="2">1ºregulation</td><td>50</td><td>66</td><td>3300</td><td>1,00e-06</td></tr>
<tr><td colspan="2">1ºregulation</td><td>100</td><td>33</td><td>3300</td><td>1,00e-06</td></tr>
<tr><td colspan="2">1ºregulation</td><td>150</td><td>15</td><td>3000</td><td>1,00e-06</td></tr>
<tr><td colspan="2">2ºregulation</td><td>50</td><td>34</td><td>1520</td><td>1,00e-06</td></tr>
<tr><td colspan="2">2ºregulation</td><td>200</td><td>34</td><td>6800</td><td>1,00e-06</td></tr>
<tr><td colspan="2">3ºregulation</td><td>50</td><td>183</td><td>8150</td><td>1,00e-04</td></tr>
</table>

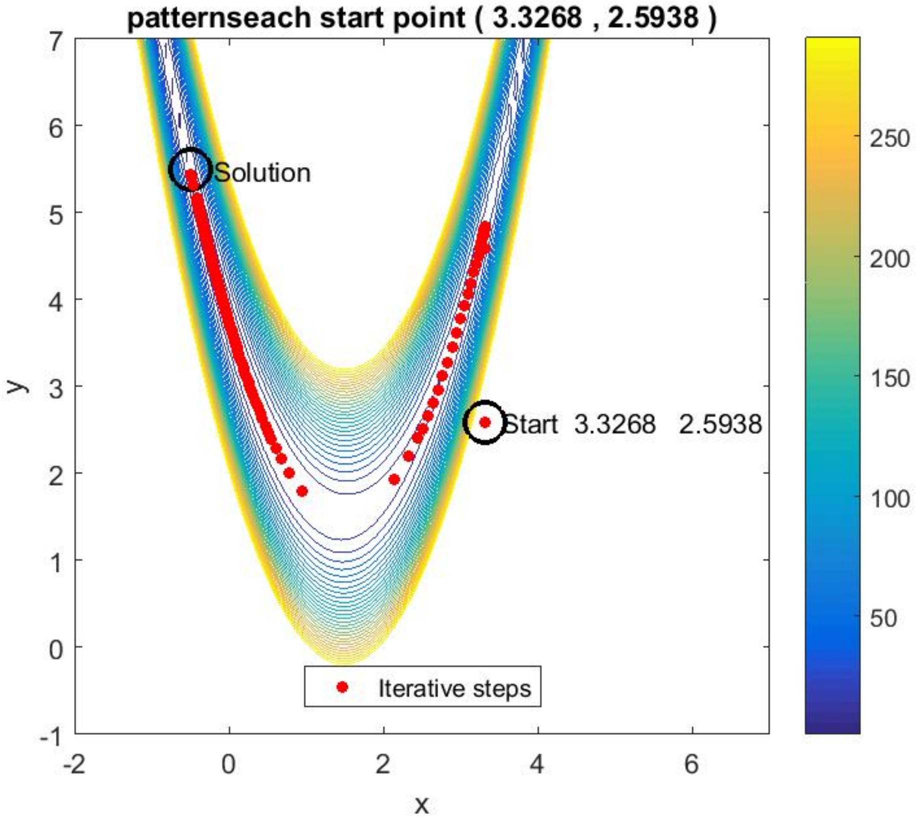
Algorithm PatterSearch

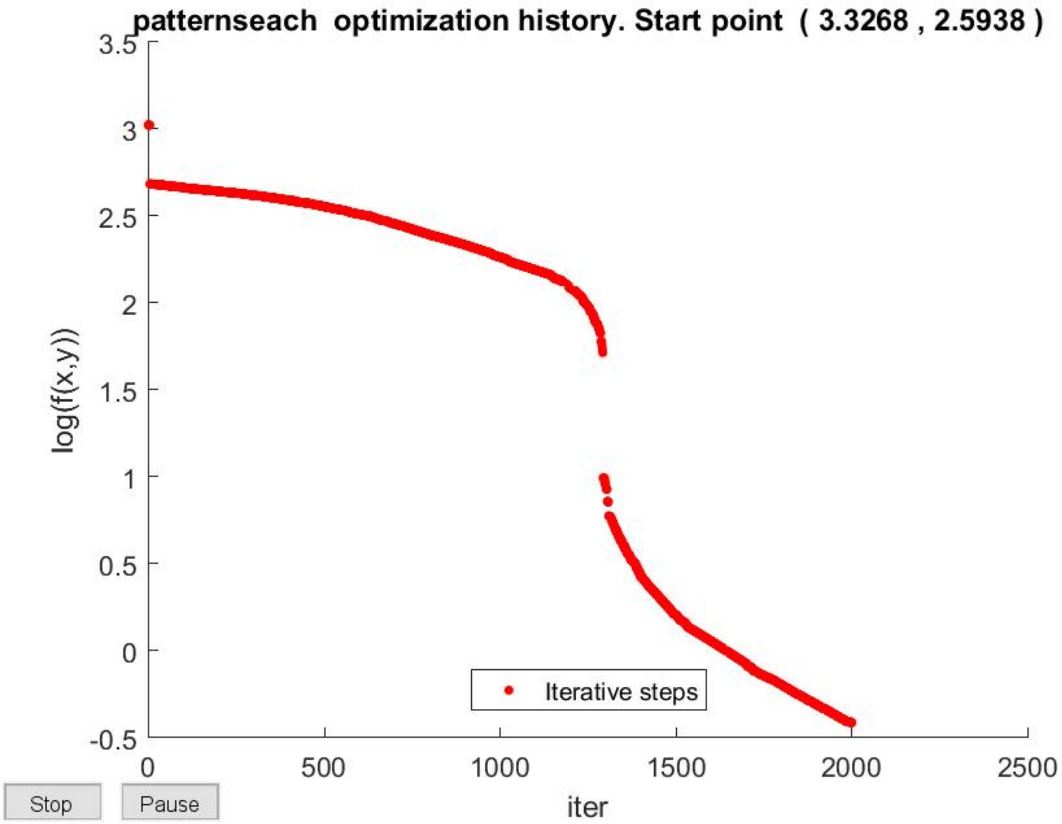
x	y	Nº f evaluations	Nº solver iterations	F min
1,753	2,056	88298	27860	7,349e-06
3,326	2,593	114440	36264	3,586e-05
2,764	3,415	49696	15262	3,057e-06
1,695	3,429	48461	14898	2,027e-06

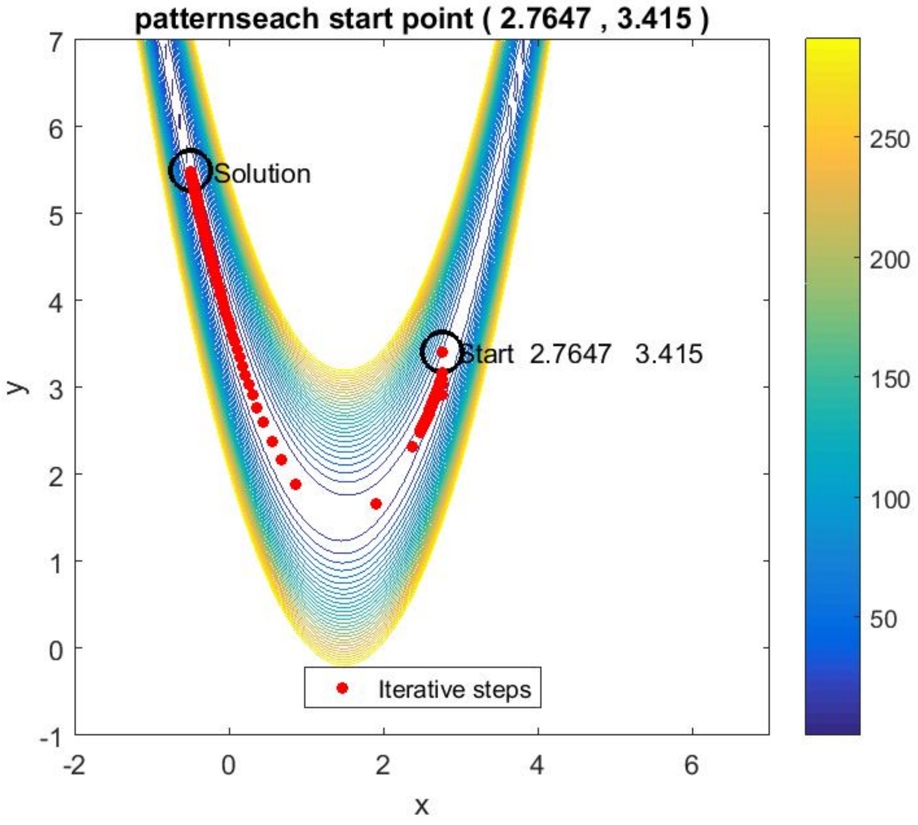
(Script OptimPatternSearch.mat)

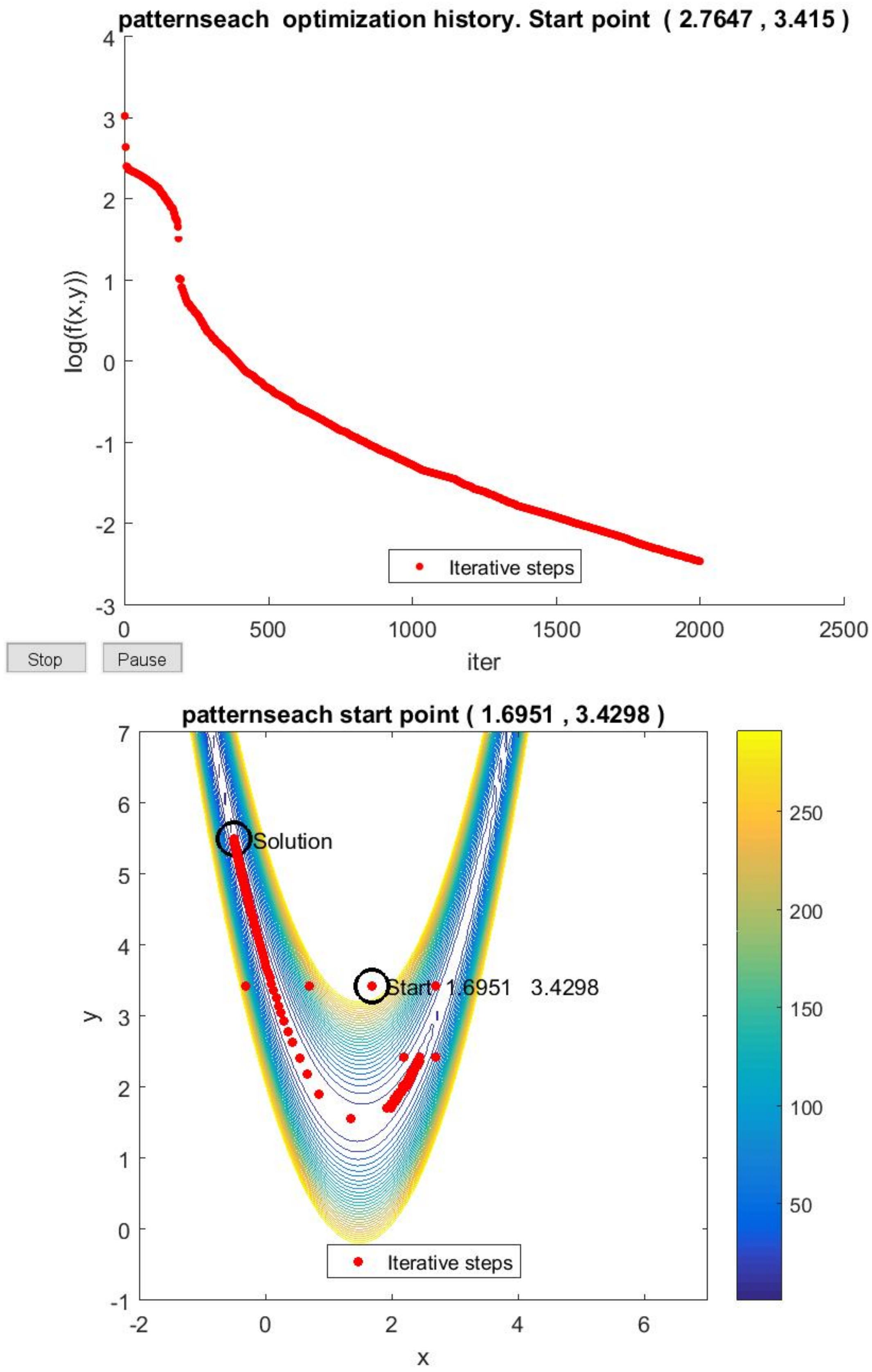

patternseach start point (1.754 , 2.057)
Solution
Start 1.754 2.057
Iterative steps
x
y
-2
0
2
4
6
-1
0
1
2
3
4
5
6
7
50
100
150
200
250

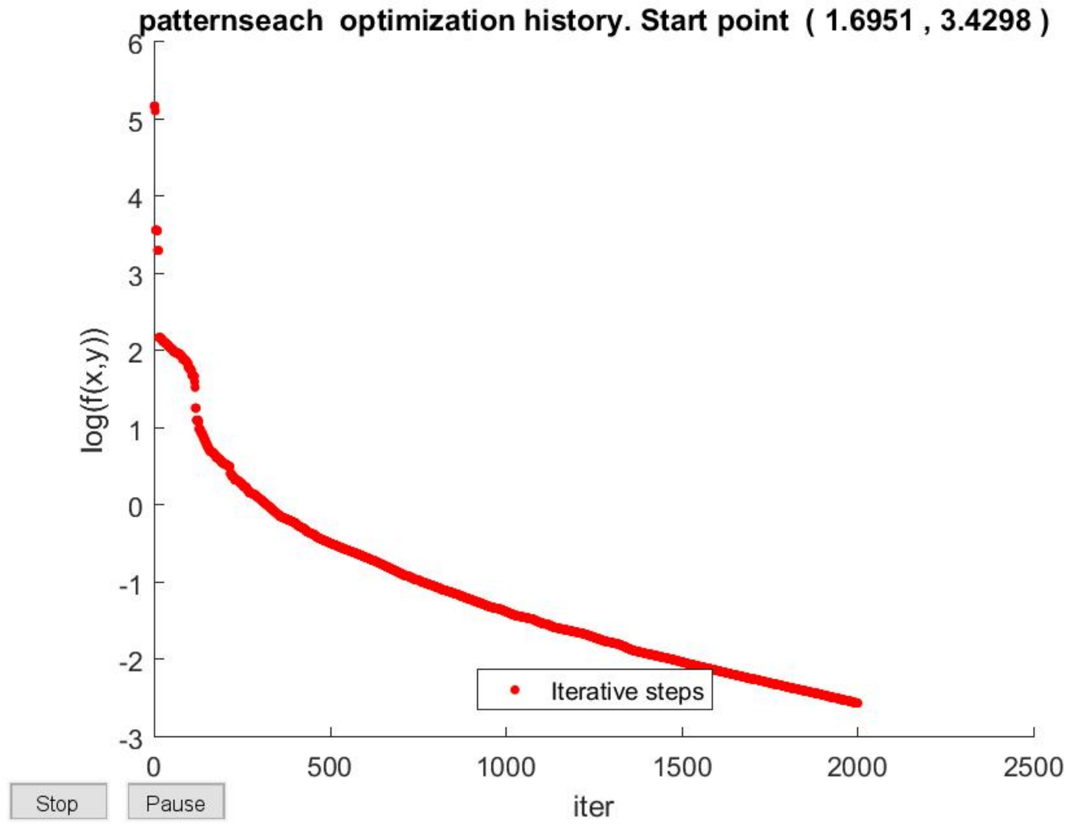

patternseach optimization history. Start point (1.754 , 2.057)
log(f(x,y))
4
3
2
1
0
-1
-2
0
500
1000
1500
2000
2500
iter
Iterative steps
Stop
Pause


patternseach start point (3.3268 , 2.5938)
Solution
Start 3.3268 2.5938
Iterative steps
y
x
7
6
5
4
3
2
1
0
-1
-2
0
2
4
6
250
200
150
100
50


patternseach optimization history. Start point (3.3268 , 2.5938)
log(f(x,y))
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
500
1000
1500
2000
2500
iter
● Iterative steps
Stop
Pause


patternseach start point (2.7647 , 3.415)
Solution
Start 2.7647 3.415
● Iterative steps
y
7
6
5
4
3
2
1
0
-1
-2
0
2
4
6
x
250
200
150
100
50


patternseach optimization history. Start point (2.7647 , 3.415)
log(f(x,y))
4
3
2
1
0
-1
-2
-3
0
500
1000
1500
2000
2500
iter
● Iterative steps
Stop
Pause
patternseach start point (1.6951 , 3.4298)
Solution
Start 1.6951 3.4298
y
7
6
5
4
3
2
1
0
-1
-2
0
2
4
6
x
● Iterative steps
250
200
150
100
50



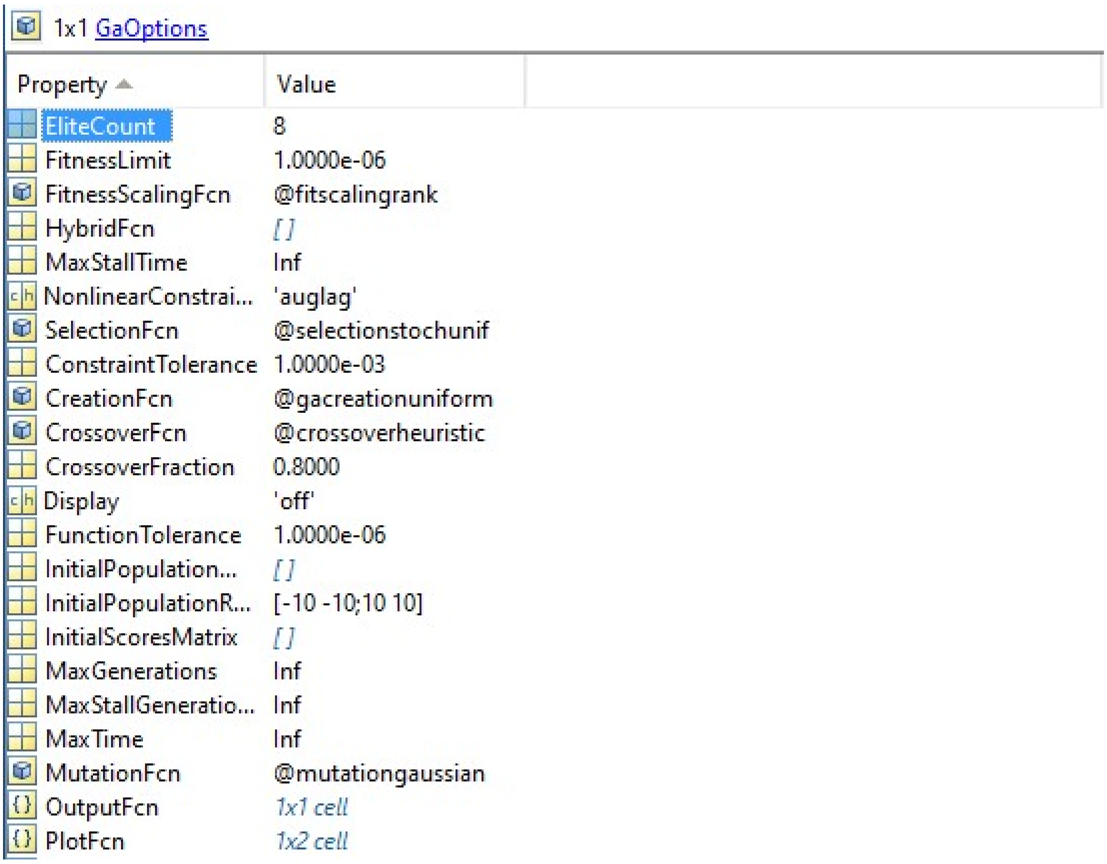
In the previous bananaOut Plots aren´t there drawn every iteration due to the heaviness of that operation, in addition for the optimization history it is just drawn till 2000 iteration for the same reason.

GA

Parameters	PopulationSize	Nº solver iterations	Total	0
1ºregulation	50	83	4015	0.00001
1ºregulation	50	66	3300	0.00001
1ºregulation	100	33	3300	0.00001
1ºregulation	150	15	3000	0.00001
2ºregulation	50	34	1520	0.00001
2ºregulation	200	34	6800	0.00001
3ºregulation	50	183	8150	0.001

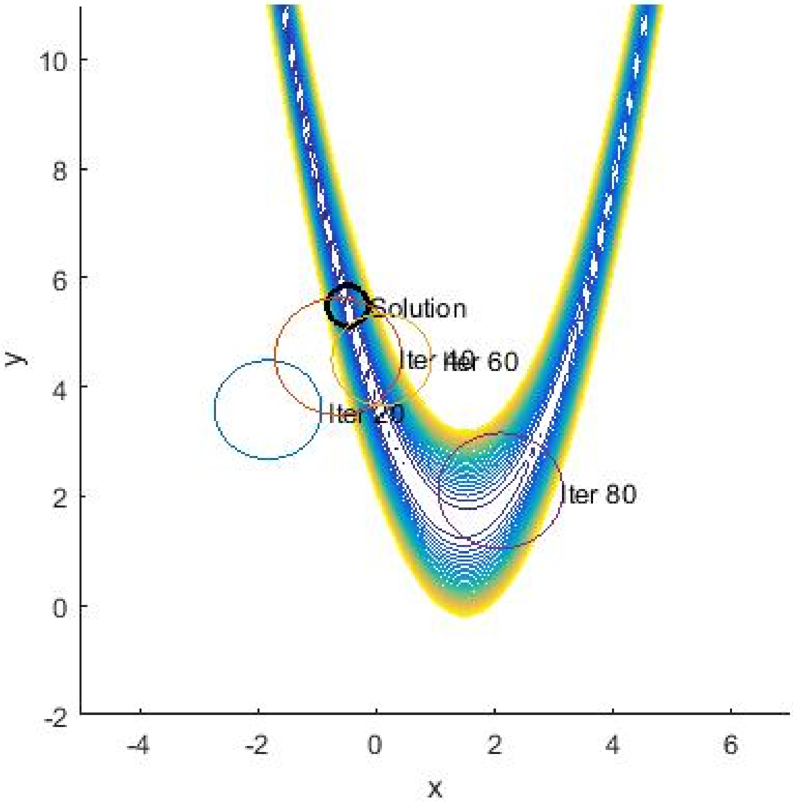
(datas optained using GA in Optimtool)

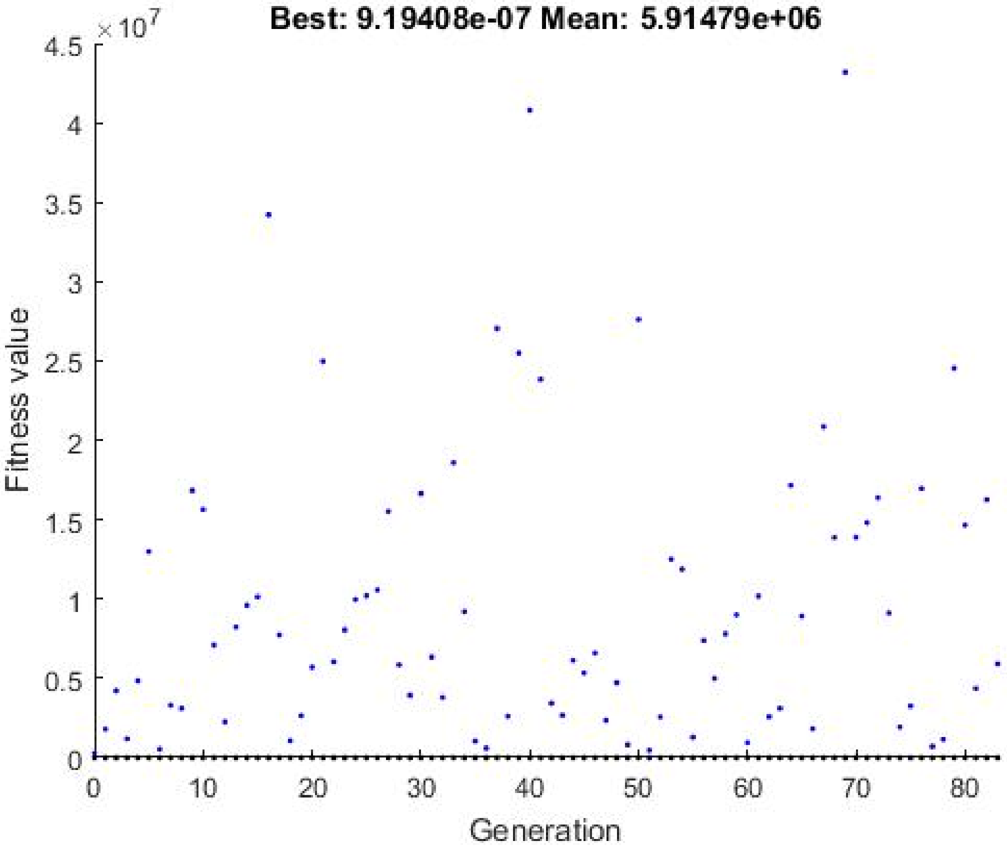
1ºregulation:

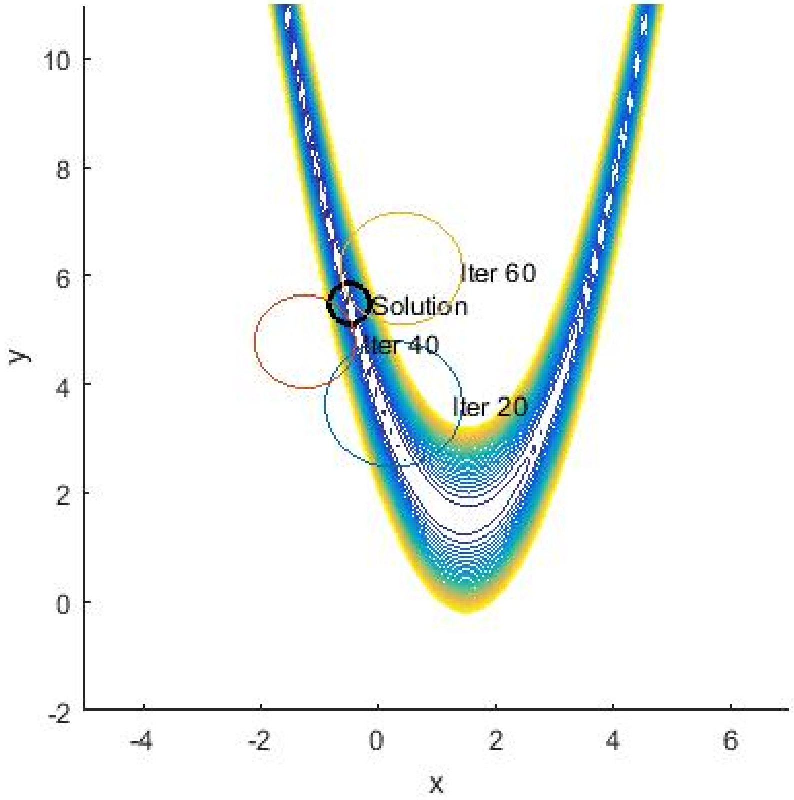


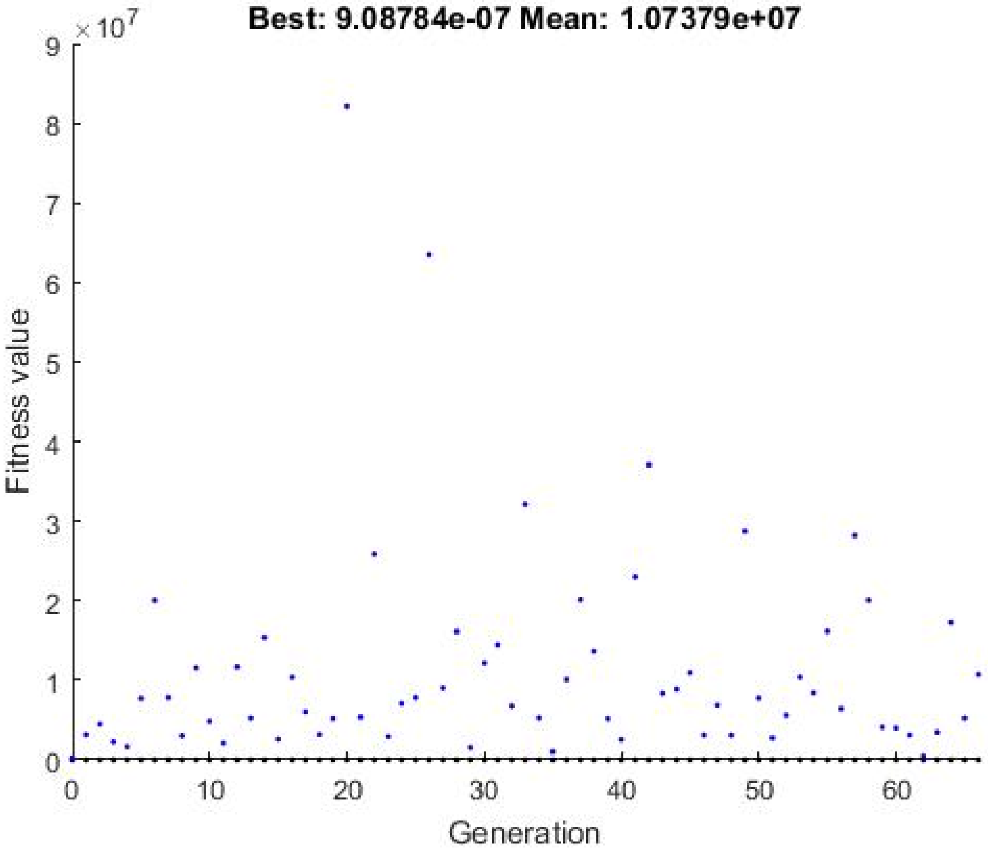
1x1 GaOptions

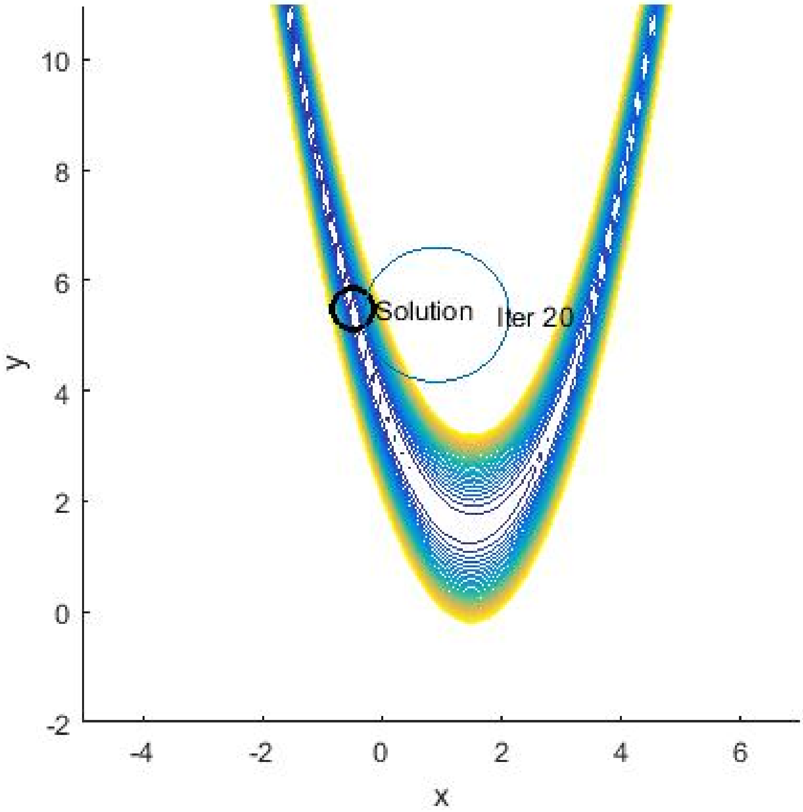
Property	Value
EliteCount	8
FitnessLimit	1.0000e-06
FitnessScalingFcn	@fitscalingrank
HybridFcn	[ ]
MaxStallTime	Inf
NonlinearConstrai...	'auglag'
SelectionFcn	@selectionstochunif
ConstraintTolerance	1.0000e-03
CreationFcn	@gacreationuniform
CrossoverFcn	@crossoverheuristic
CrossoverFraction	0.8000
Display	'off'
FunctionTolerance	1.0000e-06
InitialPopulation...	[ ]
InitialPopulationR...	[-10 -10;10 10]
InitialScoresMatrix	[ ]
MaxGenerations	Inf
MaxStallGeneratio...	Inf
MaxTime	Inf
MutationFcn	@mutationgaussian
OutputFcn	1x1 cell
PlotFcn	1x2 cell

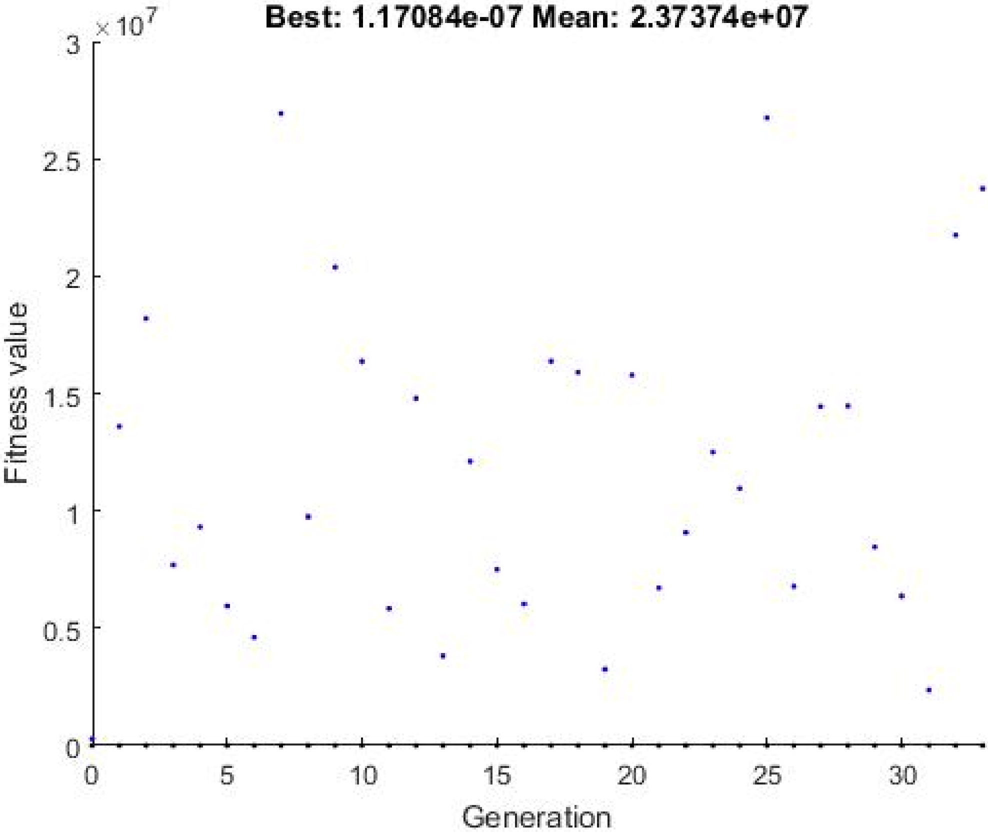

Solution
Iter 40
Iter 60
Iter 20
Iter 80
y
x
10
8
6
4
2
0
-2
-4
-2
0
2
4
6

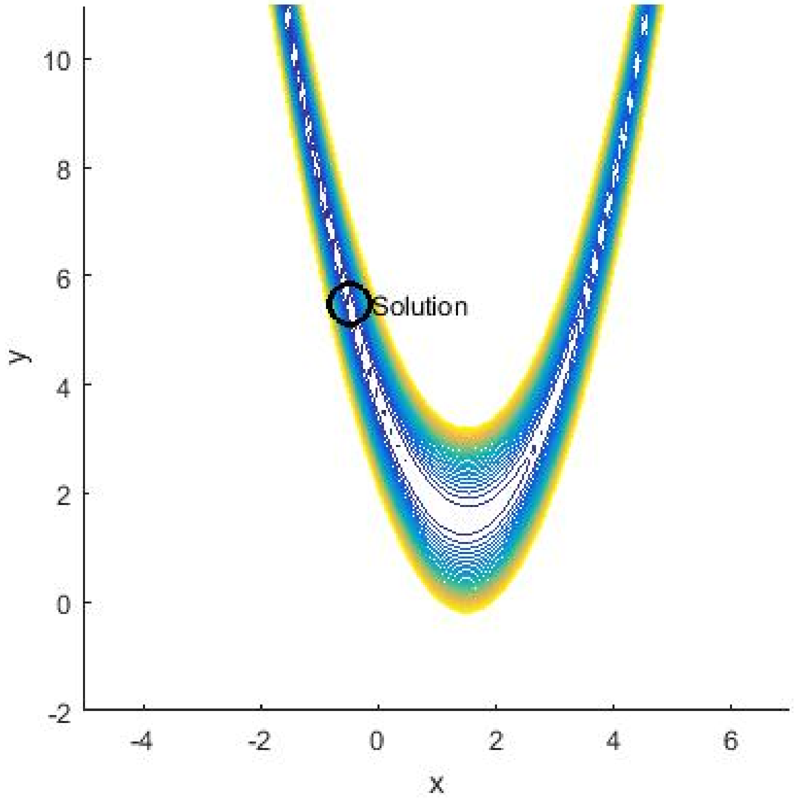

Best: 9.19408e-07 Mean: 5.91479e+06
× 10^7
Fitness value
Generation
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
10
20
30
40
50
60
70
80

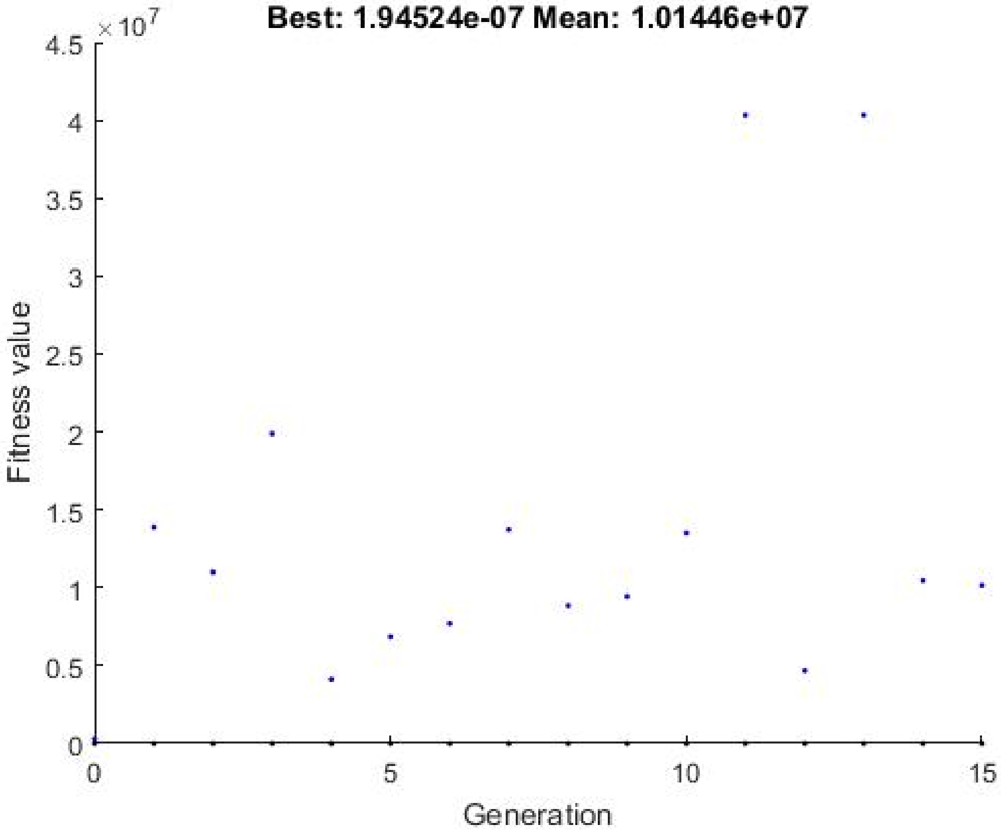

Iter 60
Solution
Iter 40
Iter 20
y
x
10
8
6
4
2
0
-2
-4
-2
0
2
4
6


Best: 9.08784e-07 Mean: 1.07379e+07
$\times 10^7$
Fitness value
Generation
9
8
7
6
5
4
3
2
1
0
0
10
20
30
40
50
60

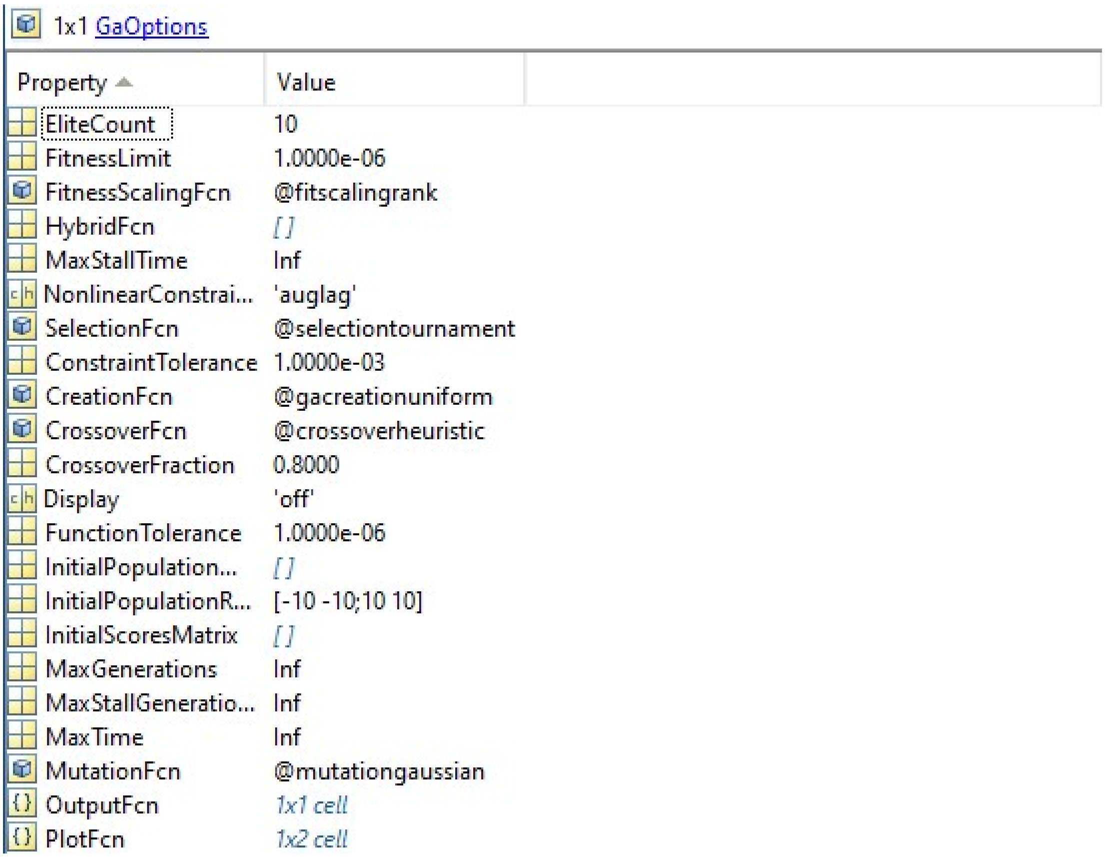

Solution
Iter 20
y
10
8
6
4
2
0
-2
-4
-2
0
2
4
6
x


Best: 1.17084e-07 Mean: 2.37374e+07
$\times 10^7$
3
2.5
2
1.5
1
0.5
0
Fitness value
0
5
10
15
20
25
30
Generation



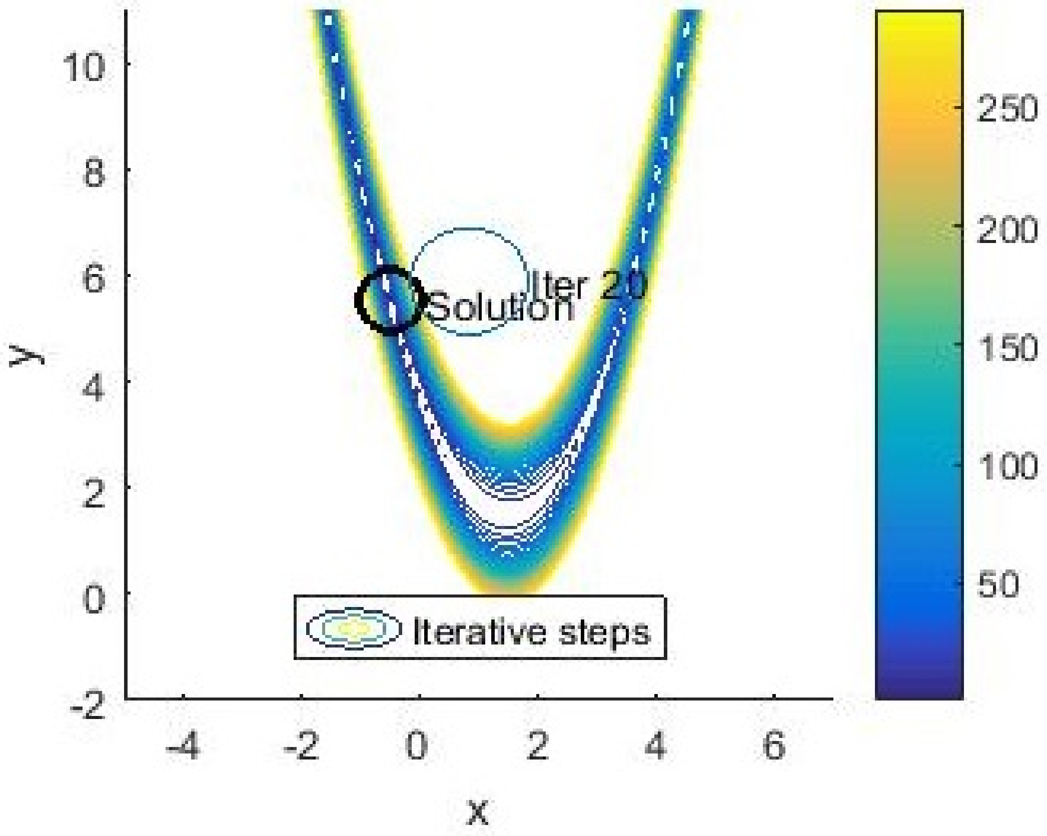


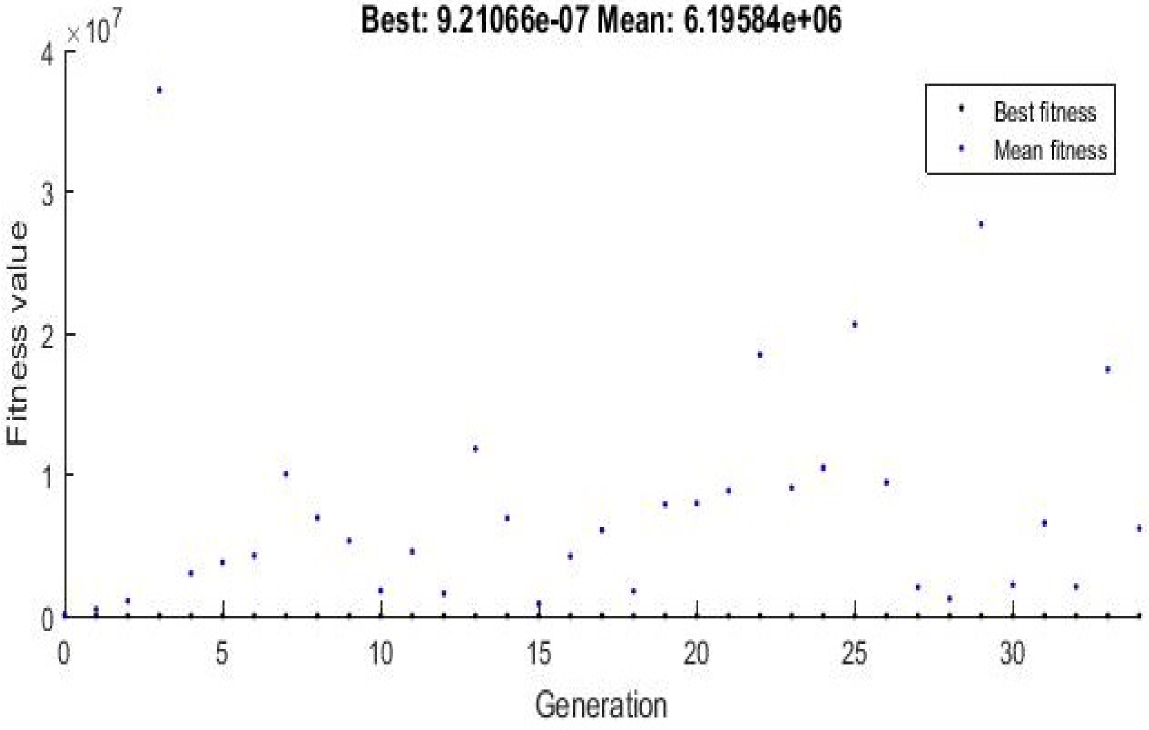
2ºRegulation

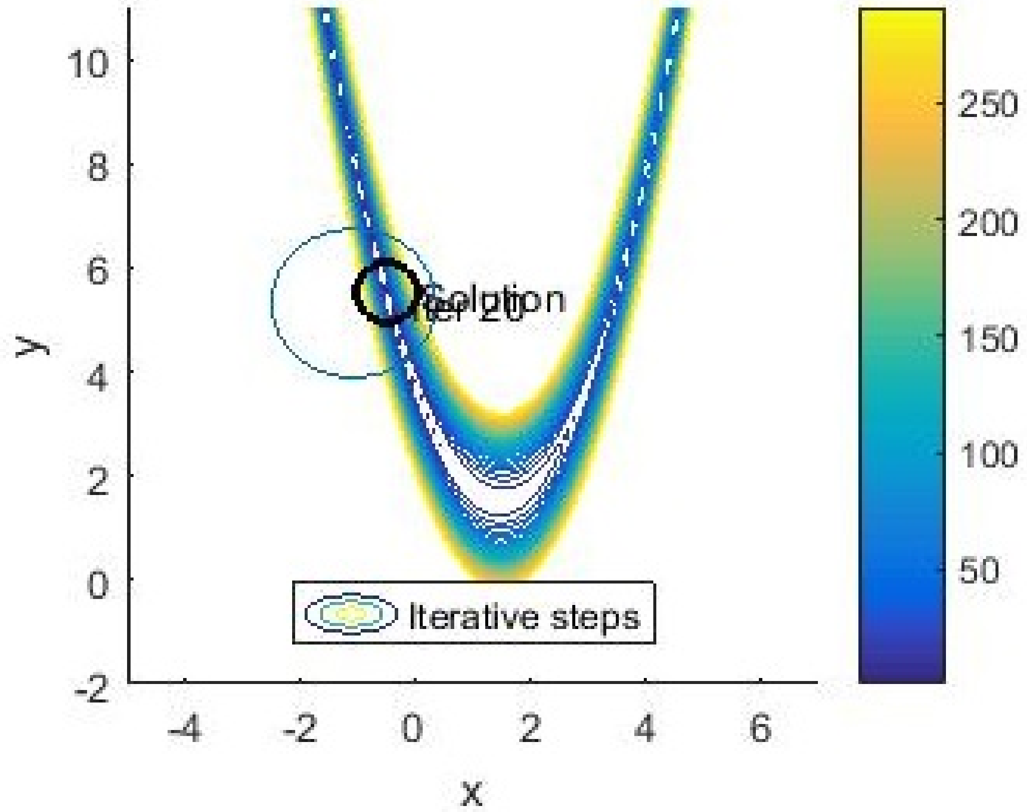


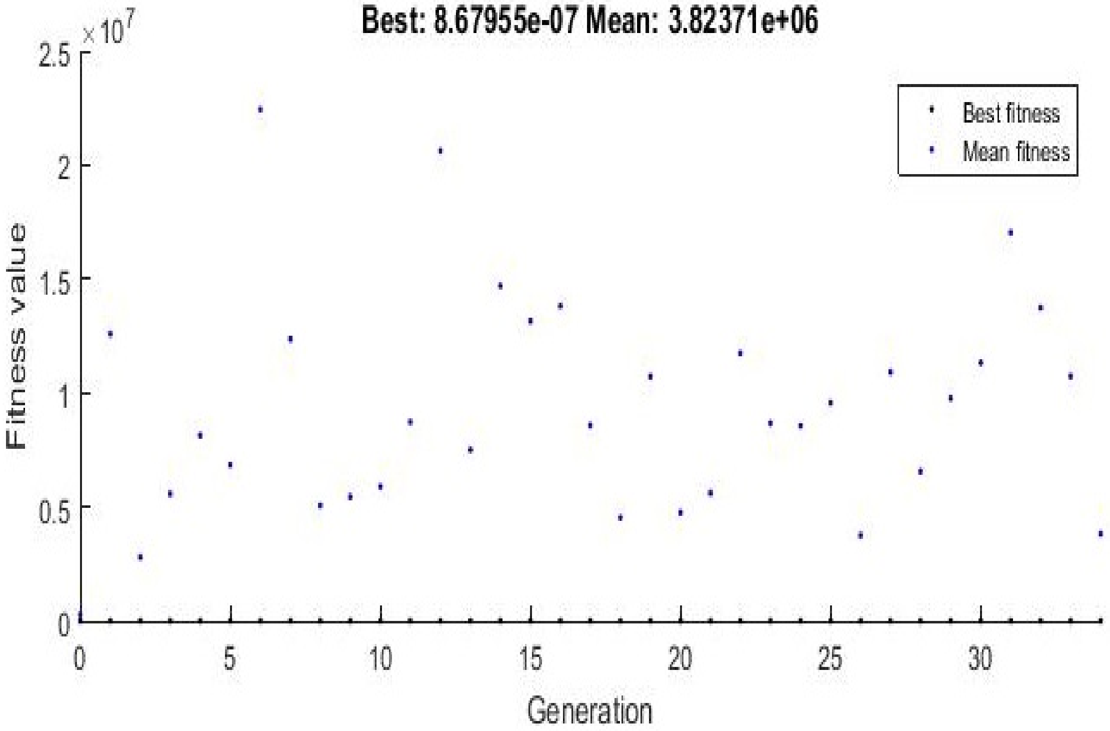
1x1 GaOptions

Property	Value
EliteCount	10
FitnessLimit	1.0000e-06
FitnessScalingFcn	@fitscalingrank
HybridFcn	[ ]
MaxStallTime	Inf
NonlinearConstrai...	'auglag'
SelectionFcn	@selectiontournament
ConstraintTolerance	1.0000e-03
CreationFcn	@gacreationuniform
CrossoverFcn	@crossoverheuristic
CrossoverFraction	0.8000
Display	'off'
FunctionTolerance	1.0000e-06
InitialPopulation...	[ ]
InitialPopulationR...	[-10 -10;10 10]
InitialScoresMatrix	[ ]
MaxGenerations	Inf
MaxStallGeneratio...	Inf
MaxTime	Inf
MutationFcn	@mutationgaussian
OutputFcn	1x1 cell
PlotFcn	1x2 cell

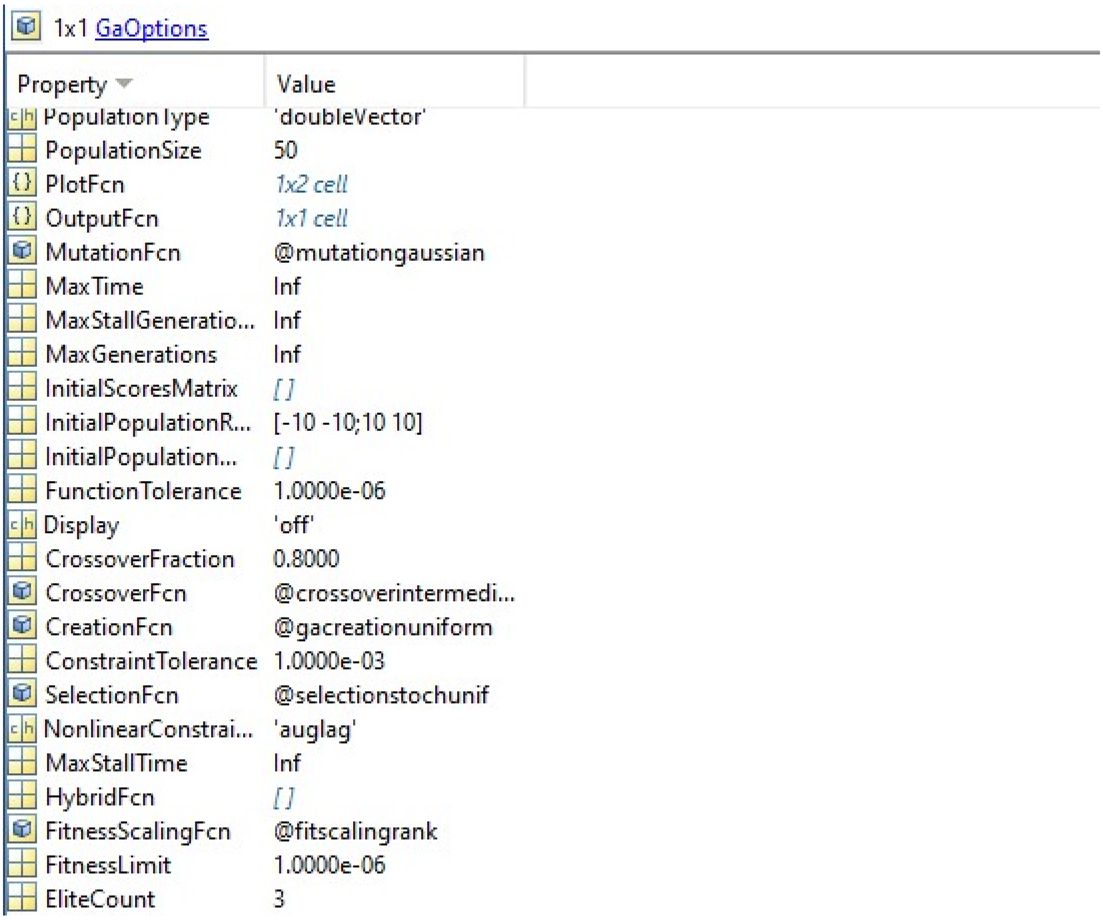

10
8
6
4
2
0
-2
y
-4
-2
0
2
4
6
x
Solution
Iter 20
Iterative steps
250
200
150
100
50


Best: 9.21066e-07 Mean: 6.19584e+06
$\times 10^7$
4
3
2
1
0
Fitness value
0
5
10
15
20
25
30
Generation
Best fitness
Mean fitness


y
x
10
8
6
4
2
0
-2
-4
-2
0
2
4
6
Solution
Iter 20
Iterative steps
250
200
150
100
50

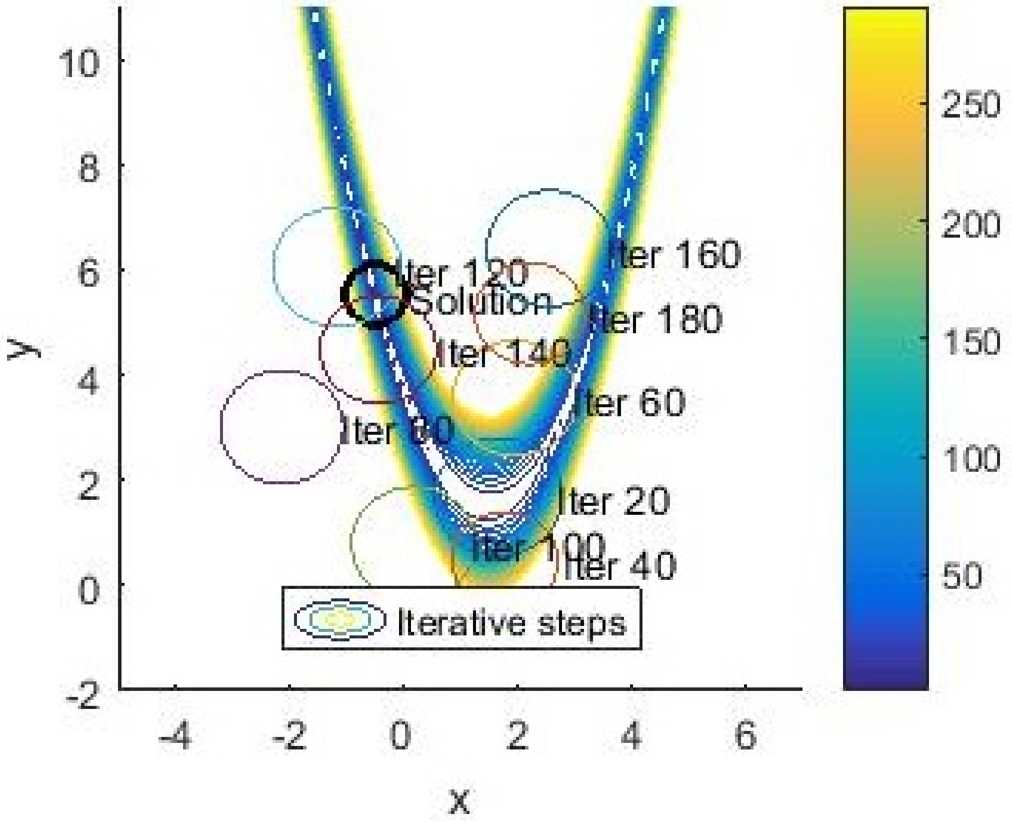

Best: 8.67955e-07 Mean: 3.82371e+06
×10^7
2.5
2
1.5
1
0.5
0
Fitness value
0
5
10
15
20
25
30
Generation
Best fitness
Mean fitness

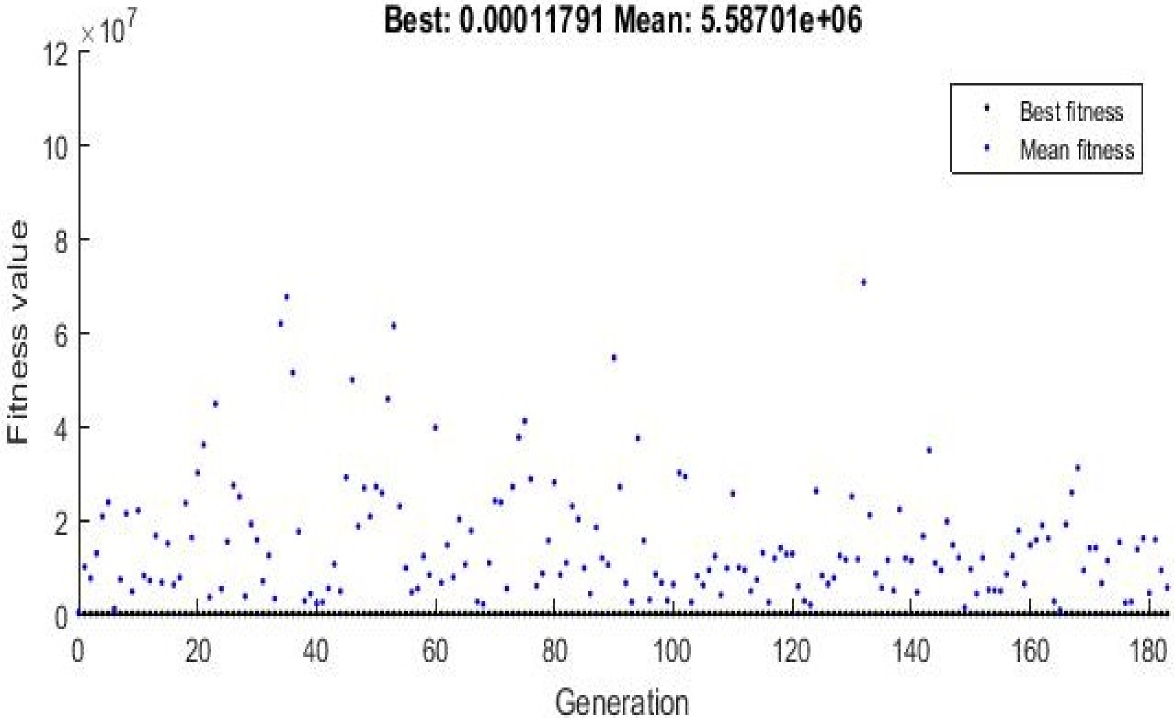
3ºRegulation



1x1 GaOptions

Property	Value
PopulationType	'doubleVector'
PopulationSize	50
PlotFcn	*1x2 cell*
OutputFcn	*1x1 cell*
MutationFcn	@mutationgaussian
MaxTime	Inf
MaxStallGeneratio...	Inf
MaxGenerations	Inf
InitialScoresMatrix	*[ ]*
InitialPopulationR...	[-10 -10;10 10]
InitialPopulation...	*[ ]*
FunctionTolerance	1.0000e-06
Display	'off'
CrossoverFraction	0.8000
CrossoverFcn	@crossoverintermedi...
CreationFcn	@gacreationuniform
ConstraintTolerance	1.0000e-03
SelectionFcn	@selectionstochunif
NonlinearConstrai...	'auglag'
MaxStallTime	Inf
HybridFcn	*[ ]*
FitnessScalingFcn	@fitscalingrank
FitnessLimit	1.0000e-06
EliteCount	3


y
x
-4
-2
0
2
4
6
-2
0
2
4
6
8
10
250
200
150
100
50
Iter 20
Iter 40
Iter 60
Iter 80
Iter 100
Iter 120
Iter 140
Iter 160
Iter 180
Solution
Iterative steps


Best: 0.00011791 Mean: 5.58701e+06
×10^7
Fitness value
Generation
0
2
4
6
8
10
12
0
20
40
60
80
100
120
140
160
180
Best fitness
Mean fitness

Conclusion:

How the results show, genetic algorithmic is more effective in this case. Also, it is true that to get that efficient I tried several times with different configuration, and with some configurations it never found a proper solution. So it means, that if you don´t know the optimum solution it could be a really difficult tast to set the configuration of a GA. On the other hand, Pattern search is a numerical algorithm does not require the calculation of gradrient vector, it use direct search technic instead, for that reasons their calculations are lighter than the calculation of other methods.

The gradient methods such as Newton-method, quasy newton-method and so on, spent less iteration to find the solution with the addicional cost of doing more calculations, and some of them assure convergerce.

To sum up, to optimizate a function we should choose carefully the algorithm keeping in mind the function, the charasteristics of the algorithms, and what it is more important for me speed, precission…