Stochastic Optimization Algorithm 2023 - Homework 2

Purusothaman Seenivasan; CID: pursee; Personal number: 19970421-8396

October 2023

2.1 The traveling salesman problem (TSP)

Ant system algorithm is implemented to find minimum distance for the travelling salesman problem whose location are given in input file. For the below parameters, the distance of 98.58691 is obtained at iteration 9 and ant 7734 yields that result.
Number of ants = 10000
Alpha = 1
Beta = 3.8
Rho= 0.3
tau_0= 0.1

The path that yields the best result is [1 5 18 26 25 32 50 29 47 30 49 37 40 44 39 27 45 38 33 48 28 46 34 42 36 8 16 21 19 7 14 22 17 24 23 9 10 13 15 6 12 2 4 20 11 3 41 35 43 31] and the plot of best path is showed in Figure 1

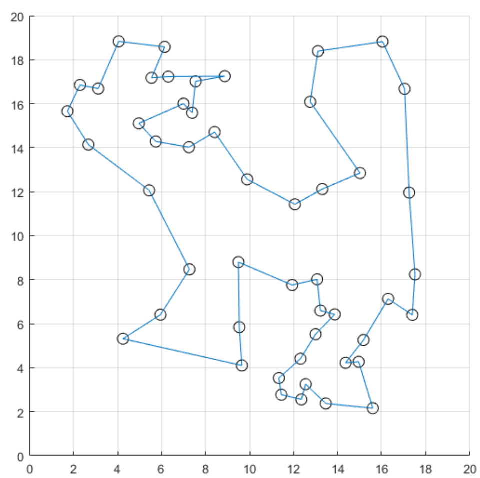

Figure 1: Plot of best path; distance=98.58691

2.2 Particle swarm optimization (PSO)

Particle Swarm optimization algorithm is implemented to find the minimum of the given function

$$f(x_1, x_2) = (x_1 + x_2 – 11)^2 + (x_1 + x_2 – 7)^2 \tag{1}$$

As mentioned in problem description, a contour plot of $log(0.01 + f(x_1, x_2))$ is created using Matlab as in Figure 2. The purpose of using $log(0.01 + f(x_1, x_2))$ in this plot is to enhance the visualization of a wide range of values for x_1 and x_2 within the given function.

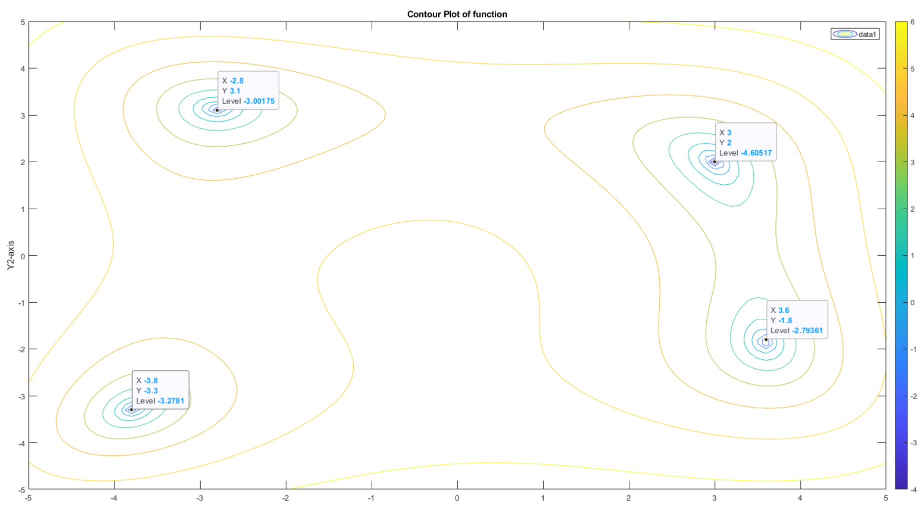

Figure 2: Contour plot of the given function and approx minimum is labelled; distance=98.58691

The code for PSA is implemented in Matlab to find the location of all local minima of the function.

x_1^*	x_2^*	$f(x_1^*, x_2^*)$
-2.8049975743	3.131692	1.182684e-06
-3.7793513305	-3.283592	6.889789e-06
3.5846037525	-1.848178	1.588316e-06
3.0000739506	2.000099	5.138508e-07

Table 1: List of x_1^* and x_2^* and its corresponding function value

To initialize position the following equation is used

$$x_{ij} = x_{min} + r(x_{max} - x_{min}) \quad i = 1..number\ of\ swarm \quad j = 1...number\ of\ variables \tag{2}$$

To initialize velocity for each element in swarm. As in this case $x_{min} = -x_{max}$ the following equation is used

$$v_{ij} = \alpha(x_{min} + r(x_{max} - x_{min})) \quad i = 1..number\ of\ swarm \quad j = 1...number\ of\ variables \tag{3}$$

When evaluating if $f(x_i) < f(x_i^{pb})$ then $x_i^{pb} = x_i$
and if $f(x_i) < f(x_i^{sb})$ then $x_i^{sb} = x_i$
where x_i^{pb} = best position of each particle
x_i^{sb} = global best position
p and q are random number
c1 and c2 are constants

To update the velocity,

$$v_{ij} = \omega v_{ij} + (\frac{c_1 q(x_{ij}^{pb} - x_{ij})}{\delta T}) + (\frac{c_2 r(x_j^{sb} - x_{ij})}{\delta T}) \tag{4}$$

where
ω is inertia weight.
c_1 and c_2 are constants
q and r are uniform random number between [0,1]
δT is time step length usually
ω <1 initially for the purpose of exploration of different possible path and then decreased by a constant β=0.9 for exploitation of better path

v_{ij} is restricted such that $|v_{ij}| < v_{max}$
To update the position

$$x_{ij} = x_{ij} + v_{ij}\delta T \tag{5}$$

The above mentioned steps are repeated until function a minimum value. The implemented code stops itself after getting 4 different minimum values.

2.4 Function fitting using LGP

The code that was submitted is not fully completed. After 10000 iteration of runLGP file, a chromosome that have maximum fitness is stored in BestChromosome.m. TestLGP function is implemented but it yields very large error
Population size= 100
Constants= [1,2,3]
Number of variables= 3
Number of operators = 4 = [+ - x /]
Tournament size=3
Mutation Probability= 0.02
Tournament Probability = 0.2
Cross over probability = 0.8