Code Part

### cost.m

function result = cost(x)

result = 0;

leng = length(x);

prod = 1;

for i = 1:leng

if (x(i) < 0 || x(i) > 10)

return;

end

prod = prod \* x(i);

end

if prod < 0.75

return;

end

num1 = 0;

num2 = 2;

denum = 0;

for i = 1:leng

num1 = num1 + (cos(x(i)))^4;

num2 = num2 \* (cos(x(i)))^2;

denum = denum + i \* (x(i))^2;

end

result = -1 \* abs((num1-num2)/sqrt(denum));

### neighbour2D.m

function x = neighbor2D(x)

index = randi(length(x));

newval = x(index) + sqrt(5).\*randn(1,1);

while (newval > 10 || newval < 0)

newval = x(index) + sqrt(5).\*randn(1,1);

end

x(index) = newval;

### SA.m

% Simulated Annealing

function [solution, sbest] = SA(sinitial, Tinitial, alpha, beta, Minitial, maxiter)

M = Minitial;

T = Tinitial;

CurS = sinitial;

BestS = CurS;

CurCost = cost(CurS);

BestCost = CurCost;

solution = zeros(maxiter, 3);

sbest = zeros(maxiter, 20);

Time = 1;

while (Time <= maxiter)

[CurS\_res, CurCost\_res, BestS\_res, BestCost\_res] = Metropolis(CurS, CurCost, BestS, BestCost, T, M);

solution(Time:Time+M-1,1) = Time:Time+M-1;

solution(Time:Time+M-1,2) = CurCost\_res;

solution(Time:Time+M-1,3) = BestCost\_res;

CurS = CurS\_res(M,:);

CurCost = CurCost\_res(M);

BestS = BestS\_res(M,:);

BestCost = BestCost\_res(M);

sbest(Time:Time+M-1,:) = BestS\_res;

Time = Time + M;

T = alpha \* T; % Update T after M iterations

M = beta \* M;

end

% Metropolis

function [CurS\_res, CurCost\_res, BestS\_res, BestCost\_res] = Metropolis(CurS, CurCost, BestS, BestCost, T, M)

i = 1;

CurS\_res = zeros(M,20);

CurCost\_res = zeros(M);

BestS\_res = zeros(M,20);

BestCost\_res = zeros(M);

while (i <= M)

NewS = neighbor2D(CurS);

NewCost = cost(NewS);

DeltaCost = (NewCost - CurCost);

if (DeltaCost < 0)

CurS = NewS;

CurCost = NewCost;

if NewCost < BestCost

BestS = NewS;

BestCost = NewCost;

end

else

if (rand(1) < exp( -1. \* DeltaCost / T))

CurS = NewS;

CurCost = NewCost;

end

end

CurS\_res(i,:) = CurS;

CurCost\_res(i) = CurCost;

BestS\_res(i,:) = BestS;

BestCost\_res(i) = BestCost;

i = i + 1;

end

### main.m

Tinitial = 0.00585;

alpha = 0.99995;

repetition = 20;

r = 10.\*rand(repetition,20);

bestsolution = zeros(10000,1);

fprintf('running SA:\n[');

for i = 1: repetition

[solution, ~] = SA(r(i,:), Tinitial, alpha, 1, 1, 10000);

bestsolution = (bestsolution .\* (i-1) + solution(:, 3)) ./ i;

fprintf('%d ', solution(10000, 3));

end

fprintf(']\n');

plot(1:10000, -bestsolution, 'g:');