|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Solver 1 | | Customer 1 | Customer 2 | Customer 3 |
| Supplier 1 | Part 01 | 4.135 | 4.609 | 5.697 |
| Supplier 1 | Part 02 | 3.430 | 5.174 | 3.787 |
| Supplier 1 | Part 03 | 5.731 | 3.044 | 9.742 |
| Supplier 1 | Part 04 | 4.609 | 4.580 | 8.368 |
| Supplier 2 | Part 01 | 5.484 | 8.549 | 5.306 |
| Supplier 2 | Part 02 | 4.376 | 4.851 | 4.301 |
| Supplier 2 | Part 03 | 6.509 | 8.391 | 7.788 |
| Supplier 2 | Part 04 | 5.280 | 8.393 | 5.015 |
| Supplier 3 | Part 01 | 7.671 | 16.443 | 7.268 |
| Supplier 3 | Part 02 | 22.099 | 10.680 | 4.720 |
| Supplier 3 | Part 03 | 15.330 | 16.060 | 6.422 |
| Supplier 3 | Part 04 | 10.065 | 11.469 | 5.629 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Solver 2 | | Customer 1 | Customer 2 | Customer 3 |
| Supplier 1 | Part 01 | 4.039 | 5.201 | 5.733 |
| Supplier 1 | Part 02 | 3.464 | 3.057 | 3.776 |
| Supplier 1 | Part 03 | 5.783 | 4.586 | 9.783 |
| Supplier 1 | Part 04 | 4.517 | 5.914 | 8.299 |
| Supplier 2 | Part 01 | 5.503 | 8.588 | 5.308 |
| Supplier 2 | Part 02 | 4.260 | 4.907 | 4.289 |
| Supplier 2 | Part 03 | 6.583 | 8.353 | 7.807 |
| Supplier 2 | Part 04 | 5.253 | 8.234 | 5.008 |
| Supplier 3 | Part 01 | 7.394 | 16.683 | 7.244 |
| Supplier 3 | Part 02 | 15.925 | 10.869 | 4.722 |
| Supplier 3 | Part 03 | 13.192 | 15.746 | 6.446 |
| Supplier 3 | Part 04 | 9.083 | 11.138 | 5.608 |
| Supplier 4 | Part 01 | 6.872 | 6.054 | 9.623 |
| Supplier 4 | Part 02 | 9.517 | 5.120 | 14.993 |
| Supplier 4 | Part 03 | 13.361 | 7.298 | 9.553 |
| Supplier 4 | Part 04 | 8.057 | 6.770 | 12.725 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Solver 3 | | Customer 1 | Customer 2 | Customer 3 | Customer 4 |
| Supplier 1 | Part 01 | 4.323 | 5.155 | 5.655 | 7.900 |
| Supplier 1 | Part 02 | 3.390 | 2.986 | 3.731 | 2.583 |
| Supplier 1 | Part 03 | 5.753 | 4.627 | 9.865 | 5.199 |
| Supplier 1 | Part 04 | 4.588 | 6.178 | 8.366 | 9.936 |
| Supplier 2 | Part 01 | 5.519 | 8.604 | 5.338 | 5.841 |
| Supplier 2 | Part 02 | 4.420 | 4.720 | 4.239 | 4.649 |
| Supplier 2 | Part 03 | 6.599 | 8.544 | 7.845 | 6.965 |
| Supplier 2 | Part 04 | 5.157 | 8.537 | 4.996 | 5.484 |
| Supplier 3 | Part 01 | 8.861 | 16.228 | 7.334 | 3.874 |
| Supplier 3 | Part 02 | 30.973 | 10.552 | 4.701 | 5.427 |
| Supplier 3 | Part 03 | 23.502 | 17.516 | 6.390 | 10.998 |
| Supplier 3 | Part 04 | 12.513 | 11.238 | 5.659 | 5.643 |

clc;

clear;

close all;

params=parameter();

j=params.j;

nVar= numel(j); % Number of Decision Variables

VarSize=[1 nVar]; % Size of Decision Variables Matrix

VarMin=0.0001; % Lower Bound of Variables

VarMax=1; % Upper Bound of Variables

%% GA Parameters

%MaxIt=120; % Maximum Number of Iterations

MaxIt=250; % Maximum Number of Iterations

%MaxIt=180; % Maximum Number of Iterations

nPop=200; % Population Size

%nPop=120; % Population Size

%nPop=150; % Population Size

%pc=0.7; % Crossover Percentage

%pc=0.8;

pc=0.9;

nc=2\*round(pc\*nPop/2); % Number of Offsprings (Parnets)

pm=0.3; % Mutation Percentage

%pm=0.2;

%pm=0.1;

nm=round(pm\*nPop); % Number of Mutants

gamma=0.05;

mu=0.9; % Mutation Rate

beta=2; % Selection Pressure

%% Initialization

empty\_individual.Position=[];

empty\_individual.Cost=[];

empty\_individual.sol=[];

pop=repmat(empty\_individual,nPop,1);

tic

for i=1:nPop

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

% Initialize Position

pop(i).Position=unifrnd(VarMin,VarMax,VarSize);

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

% Evaluation

pop(i).sol=MyCost(pop(i).Position);

pop(i).Cost= pop(i).sol.Cost;

if isnan( pop(i).Cost)==1

pop(i).Cost= pop(i).sol.Cost;

end

end

% Sort Population

Costs=[pop.Cost];

[Costs, SortOrder]=sort(Costs);

pop=pop(SortOrder);

% Store Best Solution

BestSol=pop(1);

% Array to Hold Best Cost Values

BestCost=zeros(MaxIt,1);

% Store Cost

WorstCost=pop(end).Cost;

%% Main Loop

for it=1:MaxIt

P=exp(-beta\*Costs/WorstCost);

P=P/sum(P);

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

% Crossover

popc=repmat(empty\_individual,nc/2,2);

for k=1:nc/2

% Select Parents Indices

i1=RouletteWheelSelection(P);

i2=RouletteWheelSelection(P);

if it==4

ppp=p;

i1=RouletteWheelSelection(P);

i2=RouletteWheelSelection(P);

end

% Select Parents

p1=pop(i1);

p2=pop(i2);

if it==4

p1=pop(i1);

p2=pop(i2);

end

% Apply Crossover

[popc(k,1).Position, popc(k,2).Position]=...

Crossover(p1.Position,p2.Position,gamma,VarMin,VarMax);

% Evaluate Offsprings

popc(k,1).sol=MyCost(popc(k,1).Position);

popc(k,1).Cost= popc(k,1).sol.Cost;

popc(k,2).sol=MyCost(popc(k,2).Position);

popc(k,2).Cost=popc(k,2).sol.Cost;

end

a=popc;

popc=popc(:);

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

% Mutation

popm=repmat(empty\_individual,nm,1);

for k=1:nm

% Select Parent

i=randi([1 nPop]);

p=pop(i);

% Apply Mutation

popm(k).Position=Mutate(p.Position,mu,VarMin,VarMax);

% Evaluate Mutant

popm(k).sol=MyCost(popm(k).Position);

popm(k).Cost= popm(k).sol.Cost;

end

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

% Create Merged Population

pop=[pop

popc

popm];

% Sort Population

Costs=[pop.Cost];

[Costs, SortOrder]=sort(Costs);

pop=pop(SortOrder);

% Update Worst Cost

WorstCost=max(WorstCost,pop(end).Cost);

% Truncation

pop=pop(1:nPop);

Costs=Costs(1:nPop);

% Store Best Solution Ever Found

BestSol=pop(1);

% Store Best Cost Ever Found

BestCost(it)=BestSol.sol.Cost1;

% Show Iteration Information

disp(['Iteration ' num2str(it) ' ; Best Cost = ' num2str([BestSol.sol.Cost1])

'; khata=' num2str(BestSol.sol.khata) ' ; time solution=' num2str(toc)]);

end

figure;

semilogy(BestCost,'LineWidth',2);

xlabel('it');

ylabel('F');

results.q1\_ijs=BestSol.sol.q1\_ijs;

%results.Q\_j=BestSol.sol.Q\_j;

csvwrite('ExportedData1.csv',results.q1\_ijs)

%writetable(struct2table(results),'smi.csv');