Optimize the log-periodic power law (LPPL) parameters Differential Evolution (DE) for predicting stock price crisis.

Dataset used : Daily data collected from National Stock exchange India from 2019-08-01 to 2020-05-14. Close price of N50 is used for evaluation.

Python programing language is used to Optimize the log-periodic power law (LPPL) parameters using Differential Evolution optimizer for Predicting the stock price crisis. Libraries used: scipy's optimize library,numpy,pandas,matplotlib

Differential Evolution is a stochastic, population-based optimization algorithm for solving nonlinear optimization problems. This algorithm was introduced by Storn and Price in 1996. Differential evolution try to optimize a problem by keeping a population of candidate solutions and according to its simple formulae, constructing new solutions by combining existing candidate solutions. Then keeping whichever candidate solution has the best score or fitness on the optimization problem at hand.

IMPLEMENTATION

iters : number of generations

limits : upper bound and lower bound for each fitness parameters

PopSize: Size of population

1. The above given parameters are passed to the function DE(limits,PopSize,iters)

2. Initialize mutation_factor and crossover_ratio

3. Initialize a solution vector s with s.best to be infinite. s.best contains the best fitness value and s.leader_solution contains the best candidate(leader)

Initial population

Initial population is generated randomly between upper lower and upper bound obtained from limits. calculate fitness for all the candidates in the initial population. Fitness function we use here is the LPPL fitness function . Update the s.best and s.leader_solution

Then we can start the optimization. Process is continued till the number of generations are completed or termination criteria specified by stopping_func is met.

In every iteration loop through the population and do the following steps.

1.Mutation

For each candidate solution vector A, select three other vectors except current vector randomly. Add the weighted difference of two of the vectors to the third.

d_val = population[id_1,] + mutation_factor * (population[id_2,] - population[id_3,])

d_val is called the donor vector.   mutation_factor which is predefined has to be between 0 and 1.

2. Recombination

Randomly choose a value rn and if rn> crossover_ratio then update d_val as current population candidate A. Cross over ratio can also be called recombination probability. Clip the parameters of mutant solution according to bounds.

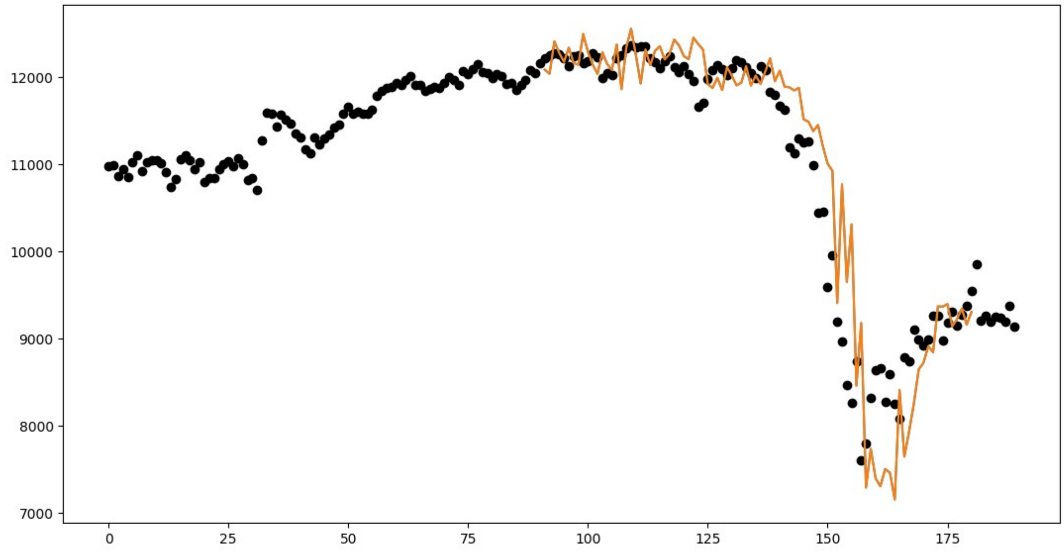
3. Selection

The current solution is compared with the mutant solution and the one with better fitness value is selected to the next generation. Also update the s.best and s.leader_solution if mutant_fitness is best. Add this solution to the list that we will use for prediction later.

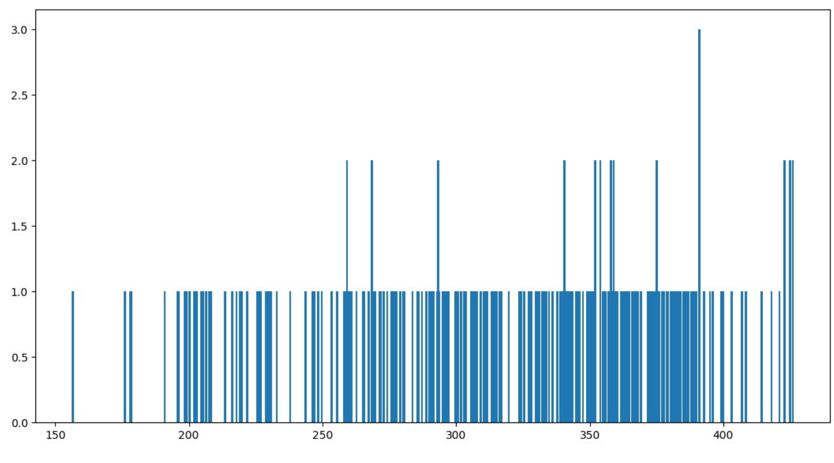
We have taken total 90 runs of DE optimization with 300 iterations in each of them.

RESULTS

Best prediction of stock prices



Critical time distribution



Critical time points


```
04714666776: 1, 342.23620941208236: 1, 255.41949649874843: 1, 325.43933502330987: 1, 260.8925835776884: 1, 342.27295491022005: 1, 310.76282347
1393: 1, 280.3205635732716: 1, 228.95093653531953: 1, 349.0510333861878: 1, 176.02075634062874: 1, 372.29634769987825: 1, 226.07385774799633:
1, 342.4366563824788: 1, 276.77784373080664: 1, 373.8405881770793: 1, 217.81334535932888: 1, 356.0625128912413: 1, 399.9022349725695: 1, 297.3
0006266618346: 1, 202.153205341343: 1, 369.18629364710705: 1, 332.3109286924463: 1, 296.91969447813005: 1, 351.34694141030366: 1, 243.63752113
19996: 1, 319.7234069633048: 1, 317.0908401023103: 1, 381.4820761029334: 1, 246.1919049488518: 1, 277.81667151827406: 1, 367.1231904098918: 1,
 300.2055274137485: 1, 276.70768609291946: 1, 271.48537420067964: 1, 313.3191150148911: 1, 358.41359773463125: 1, 355.1756830740874: 1, 382.97
609903895597: 1, 198.47431674370813: 1, 232.9140752167505: 1, 237.91517393123797: 1, 291.62665242780326: 1, 314.0445822295776: 1, 274.24221439
99158: 1, 290.95107694856426: 1, 359.4884305265281: 1, 391.0: 3, 249.72773622870417: 1, 392.8442838991949: 1, 226.00611620449837: 1, 347.45144
6580185: 1, 344.9350893759878: 1, 290.82583685466284: 1, 310.37490359683295: 1, 285.7356282616: 1, 302.852992257231: 1, 337.9065271231992: 1,
339.94471504854164: 1, 364.88184289737876: 1, 346.0878030520264: 1, 343.66763738502397: 1, 362.74284458089096: 1, 259.889805947029: 1, 293.541
051370052: 1, 355.3132687506406: 1, 257.97550064693826: 1, 319.701209025195: 1, 364.11991038601445: 1, 213.507174817512: 1, 345.50247947835885
: 1, 283.72579591520537: 1, 271.34964963182017: 1, 386.64847785936706: 1, 399.2433267918467: 1, 311.6859538701117: 1, 207.7509972984018: 1, 38
9.3844065171277: 1, 314.7139091558392: 1, 383.77091712565255: 1, 349.12929781416284: 1, 355.06754872788144: 1, 360.427046219305: 1, 342.179487
7812928: 1, 267.327071507212: 1, 330.1494594948431: 1, 292.8852995930898: 1, 259.8430626862818: 1, 205.21773562972052: 1, 334.7967995369695: 1
, 178.243370756026: 1, 219.76143742489646: 1, 371.8788325184307: 1, 265.0638913421267: 1, 307.76909164396704: 1, 327.2782477444522: 1, 200.245
63036315982: 1, 246.36116993920083: 1, 226.95040369063372: 1, 367.43718584432776: 1, 268.51494569958436: 2, 311.72185501488275: 1, 269.2628712
9676825: 1, 275.8552796189238: 1, 314.1865787010309: 1, 253.43618500587417: 1, 205.16494501014387: 1, 221.84206955930006: 1, 403.1677825506665
: 1, 417.9932551443286: 1, 225.67318711365672: 1, 219.2040964593277: 1, 290.2984315659606: 1, 269.26265293639096: 1, 267.45245515237957: 1, 36
3.1689100572547: 1, 385.1170069619835: 1, 387.9890776808513: 1, 325.53253502332245: 1, 300.85445036286745: 1, 248.37823016654198: 1, 384.0: 1,
 195.76834333437313: 1, 310.4120533663894: 1, 366.61432437110653: 1, 290.03537094300526: 1, 306.21845104855424: 1, 293.384379597028: 2, 306.95
173891101615: 1, 323.6314157070336: 1, 190.9843778555047: 1, 289.94391798103186: 1, 309.1485712871832: 1, 378.960270773264: 1, 306.68049122459
496: 1, 199.20988270217606: 1, 357.6643804465599: 1, 268.5508558334533: 1, 280.7697288438994: 1, 341.43354747516514: 1, 342.2895060732626: 1,
343.64460892361933: 1, 295.07124484803626: 1, 342.86874676097136: 1, 330.78840877979724: 1, 295.8625057402685: 1, 265.5119206192751: 1, 350.78
91426259531: 1, 388.42585574891973: 1, 288.7743730429726: 1, 324.35872373121117: 1, 341.0: 1, 316.60628275038215: 1, 346.0: 1, 260.21797848354
646: 1, 333.0370705909031: 1, 352.0: 2, 400.0: 1, 354.0: 2, 356.0: 1, 357.0: 1, 358.0: 2, 359.0: 2, 360.0: 1, 366.48702498384705: 1, 364.0: 1,
 247.11356897820085: 1, 366.0: 1, 216.1587351976351: 1, 368.0: 1, 361.65596960481946: 1, 299.90780469252513: 1, 373.0: 1, 375.0: 2, 376.0: 1,
377.0: 1, 377.7892750228119: 1, 328.11459287811385: 1, 382.0: 1, 316.1724548584239: 1, 386.0: 1, 387.0: 1, 357.44638170066446: 1, 389.0: 1, 39
0.0: 1, 345.4519199019288: 1, 388.0: 1, 395.0: 1, 396.0: 1, 360.24594505291003: 1, 381.0: 1, 379.32553635652016: 1, 260.47470650693174: 1, 301
.80954565047557: 1, 196.20664269955: 1, 300.59177277350557: 1, 277.1241220241633: 1, 229.74991042235243: 1, 293.10031879069805: 1, 285.4537124
9145455: 1, 329.9424823382186: 1, 396.073098612225: 1, 423.0: 2, 408.47214699734906: 1, 425.0: 2, 426.0: 2, 295.1880904613936: 1, 268.43442927
61734: 1, 363.19601272829027: 1, 414.30466897216525: 1, 156.58537909189928: 1, 269.7149485546744: 1, 205.15459350204145: 1, 258.6177810355309:
 1, 305.7313068209857: 1, 386.05320661973514: 1, 315.29915106736286: 1, 355.7732999178545: 1, 375.3611914215452: 1, 407.0: 1, 373.213975251638
2: 1, 303.53807219625435: 1, 336.21568503097154: 1, 380.2929561653792: 1, 208.31591181895018: 1, 296.43492455752954: 1, 262.7553592790824: 1,
204.54728789666797: 1, 324.15950346414115: 1, 272.81822744716607: 1, 339.3281235069288: 1, 333.6512356428126: 1, 279.15160239171144: 1, 178.57
091853288233: 1, 421.0: 1, 331.1913256081185: 1, 374.63475902676805: 1, 339.2416929700904: 1, 382.16771895182524: 1, 230.65643089229786: 1, 34
4.78161214139146: 1, 332.25524575186313: 1, 286.0458958106917: 1, 299.61987316875326: 1, 339.016843885194: 1, 350.0081378028108: 1, 336.105223
27379715: 1, 340.5014639288024: 2, 295.1784662989518: 1}
```