Cuckoo Search via Lévy flights

X. S. Yang and Suash Deb

NABIC, 2009, IEEE

Presented by

Cihan Kaya

What is cuckoo search with Levy flights?

- A meta-heuristic method
- Global optimization
- Based on obligate brood parasitic behavior of cuckoo birds

Wikipedia

Brood parasitism of cuckoo birds

- ❖Lay their eggs in the nest of a host bird.
- ❖Imitate the colors and patterns of host eggs.
- ❖Increase their survival and productivity.

Aidala et. al, (2010) *Nature Education Knowledge* 3(10):53

What if egg is discovered?

- Discovered foreign egg will be thrown or host will leave nest.
- Nests with eggs are selected.
- Cuckoo eggs will hatch earlier than host egg.

Aidala et. al, (2010) *Nature Education Knowledge* 3(10):53

Then what?

- ❖Cuckoo chick will evict all host eggs.
- ❖Increased food share.

Anderson et. al, (2009) *Plos One 4*(11), e7725

Laying eggs and evolutionary arm race

- Video

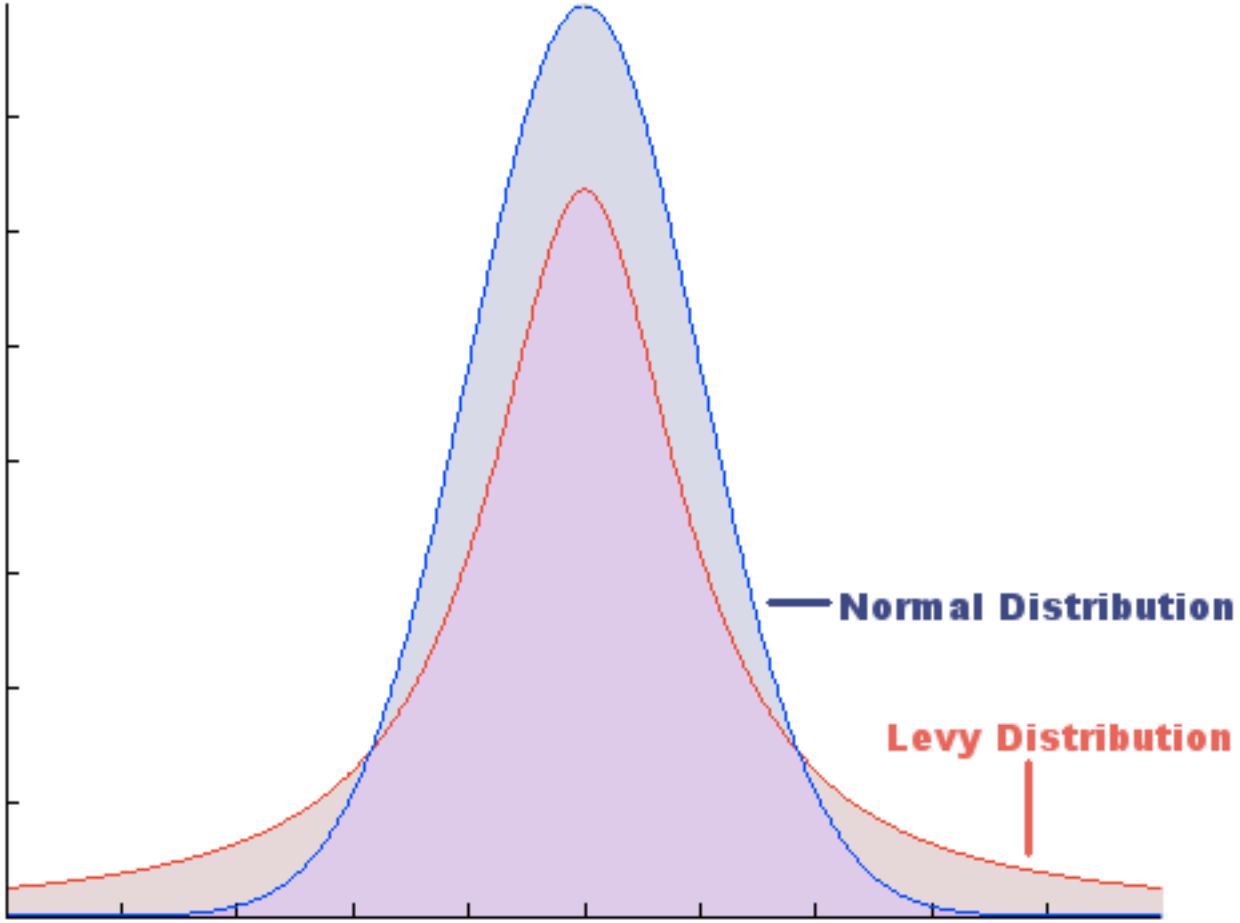
Cuckoo infiltration

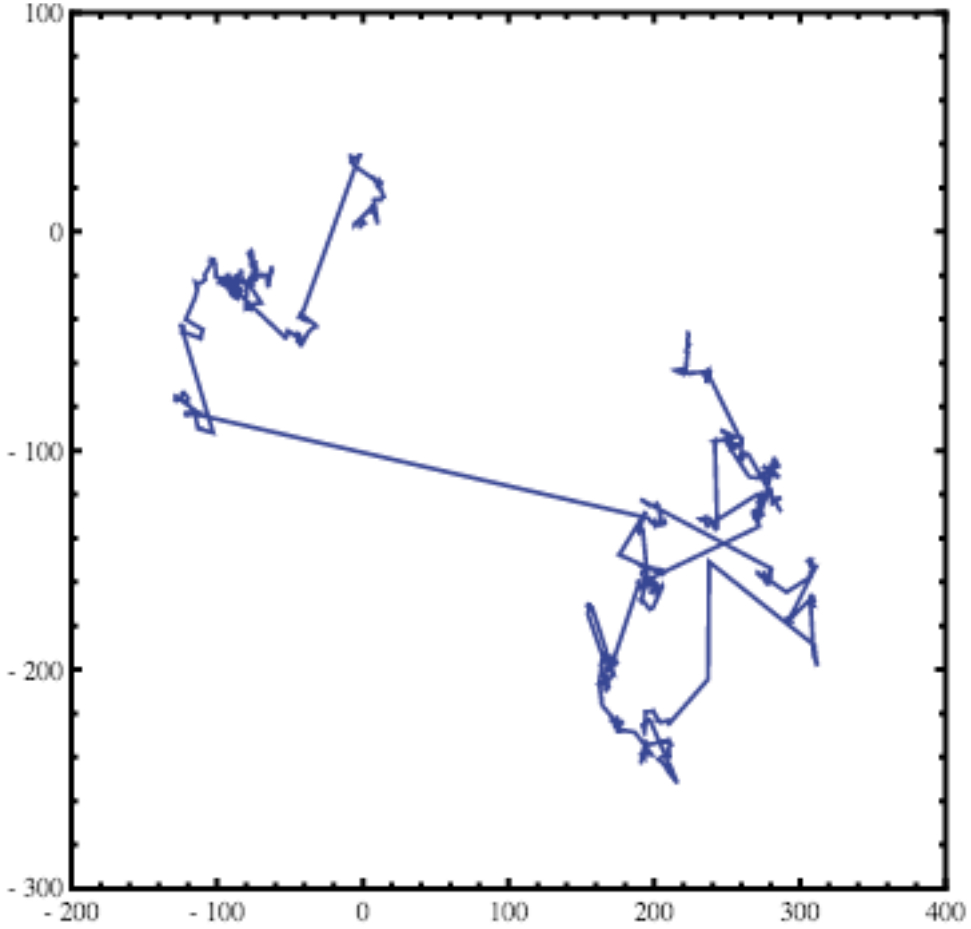
Egg destruction

Levy flights

- Food search in nature is random or quasi-random.
- Foraging path is random walk and depends on current location and transition probability.
- Since next direction is based on probability, it can be modeled mathematically.

Difference from random walk





Wikipedia

Biological inspiration

- Eggs in nests : set of solutions
- Cuckoo egg : new solution.

- New and better solutions will replace, less fit solutions.

- Cuckoo’s change position with Levy flights after leaving nest.

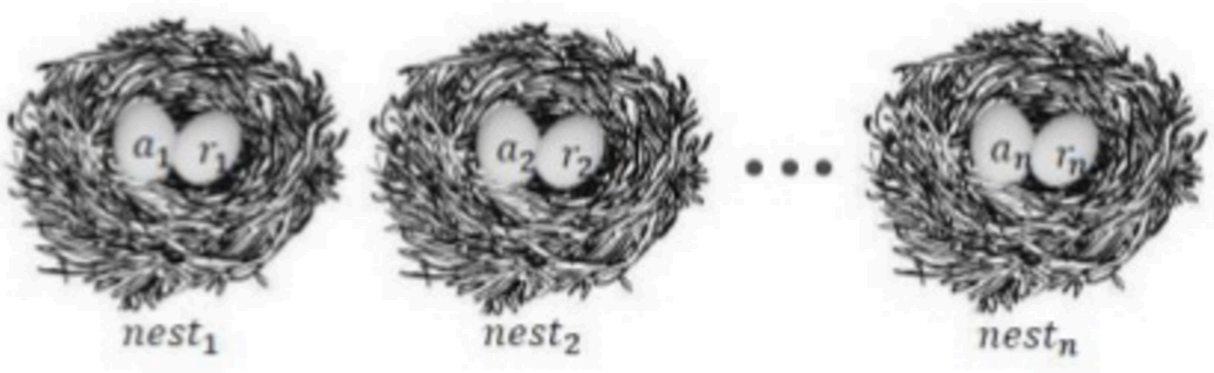
Rules of implementation

- Each cuckoo can lay one egg at each time step.
- High quality nests will carry onto next generations.
- # of host nests is fixed and p_a is the probability of discovery of an alien egg.
- Host bird can throw away egg or leave nest.

Initialization

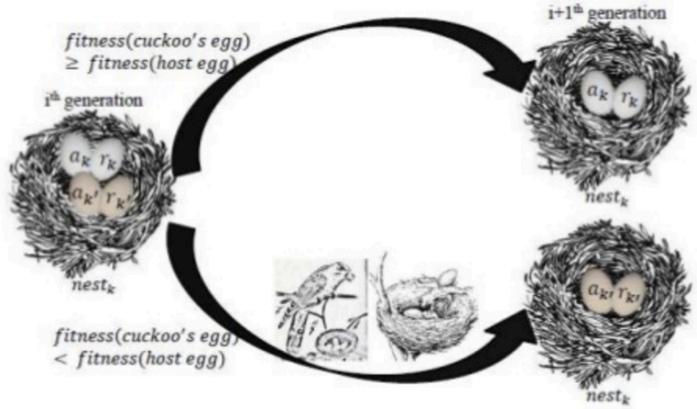
- Parameters
  - n : number of host nests
  - p_a : probability of discovery of alien egg
  - MaxIter : maximum number of iterations

- Initialization
  - Generate initial n host, $x_i^{(t)}$
  - Evaluate $f(x_i^{(t)})$



Iterations

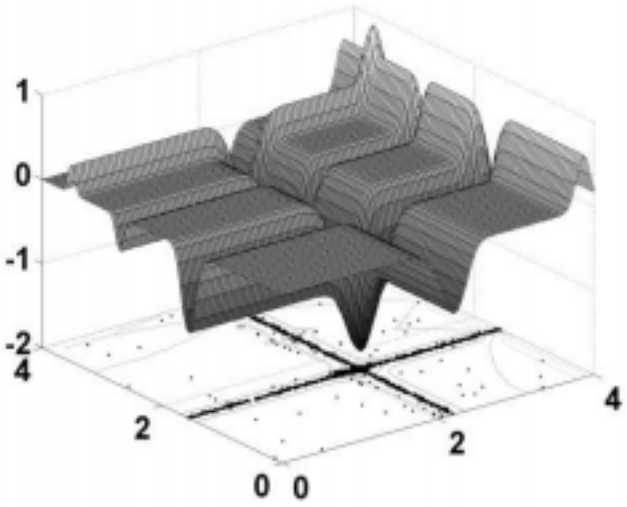
- Generate a new solution
  - $x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Le'vy(\lambda)$
  - Evaluate $f(x_i^{(t+1)})$
- Choose a nest x_j randomly
  - If $f(x_j^{(t)}) < f(x_i^{(t+1)})$
    - Replace $x_j^{(t)}$ with $x_i^{(t+1)}$
- Abandon a fraction of p_a worse nests.
  - Build new nests with Levy flights
  - Keep the best solutions

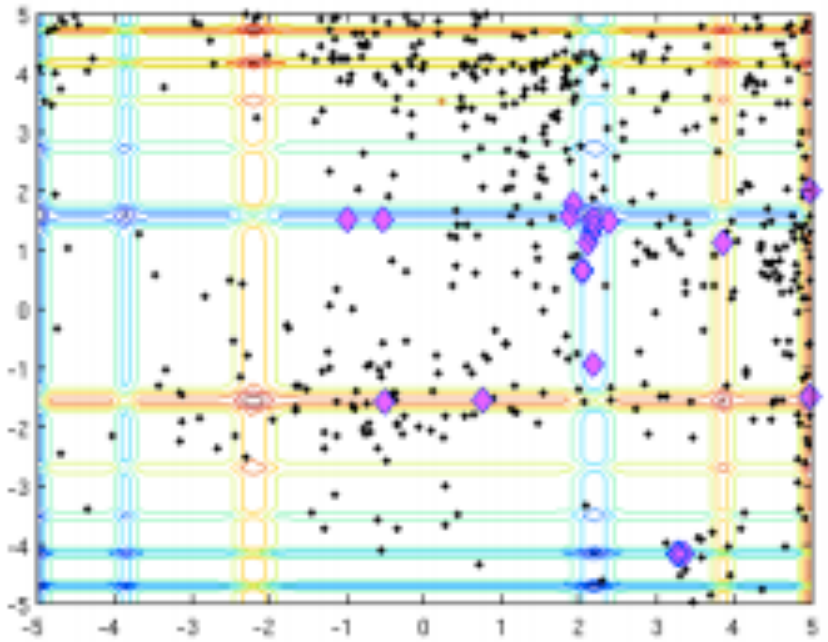


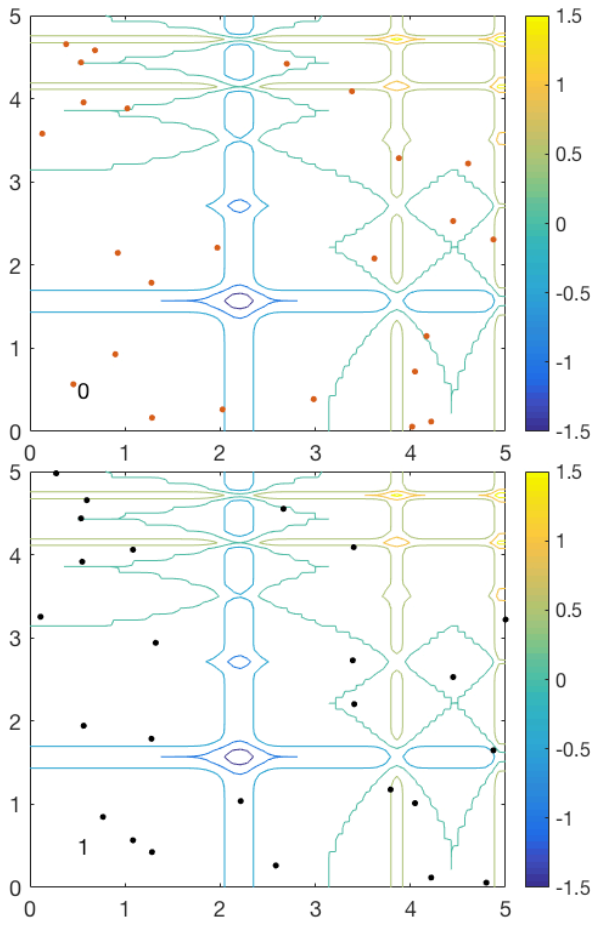
Realisation and Verification

❖Bivariate Michaelwicz function

$$f(x,y) = -\sin(x)\, sin^{2m}\left(\frac{x^2}{\pi}\right) - \sin(y)\, sin^{2m}\left(\frac{2y^2}{\pi}\right)$$



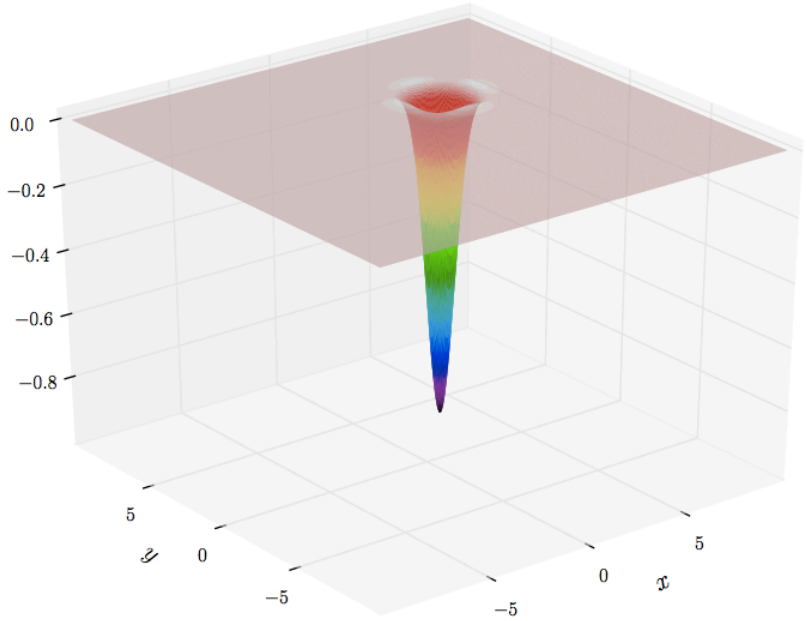


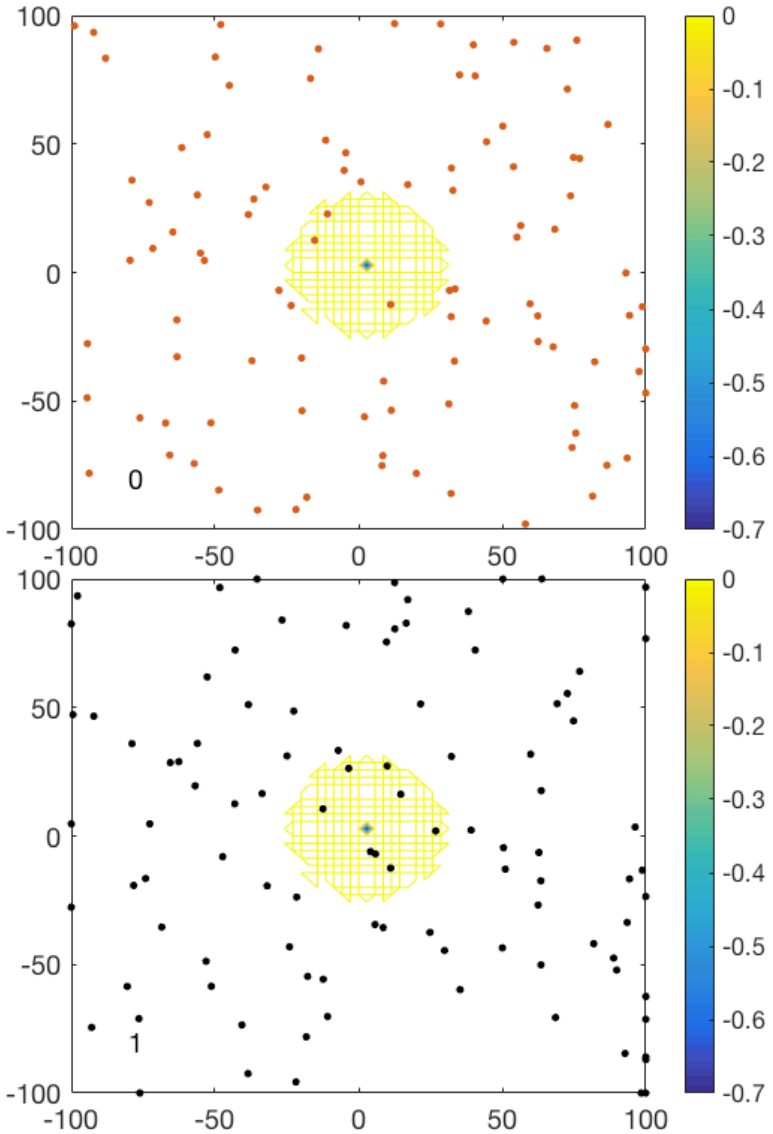


Realisation and Verification

- Easom Test Function

$$f(x,y) = -\cos(x)\cos(y)\exp\left[-(x-\pi)^2 - (y-\pi)^2\right],$$





Comparison with other algorithms

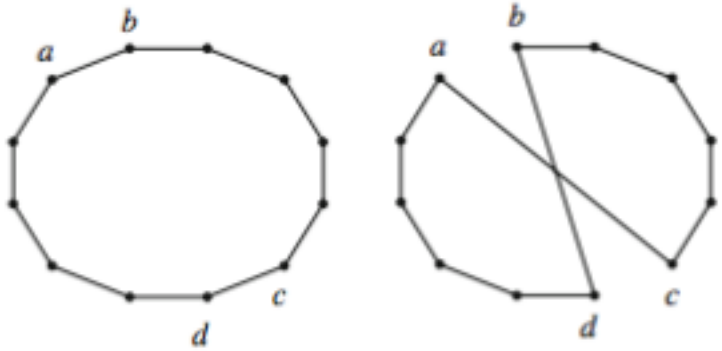
Algorithms /Functionss	Multiple Peaks	Michalewiez's (d=16)	Rosenbrock's (d=16)	De Jong's (d=256)	Schwefel's (d=128)	Ackley's (d=128)	Rastrigin's	Easom's	Griewank's	Shubert's (18 minima)
GA	52124± 3277 (98%)	89325± 7914 (95%)	55732± 8901 (90%)	25412± 1237 (100%)	227329 ±7572 (95%)	32720± 3327 (90%)	110523 ±5199 (77%)	19239± 3307 (92%)	70925± 7652 (90%)	54077± 4997 (89%)
PSO	3719± 205 (97%)	6922± 537 (98%)	32756± 5325 (98%)	17040± 1123(100%)	14522± 1275 (97%)	23407± 4325 (92%)	79491± 3715 (90%)	17273± 2929 (90%)	55970± 4223 (92%)	23992± 37557 (92%)
CS	927± 105 (100%)	3221± 519 (100%)	5923± 1937 (100%)	4971± 754 (100%)	8829± 625 (100%)	4936± 903 (100%)	10354± 3755 (100%)	6751± 1902 (100%)	10912± 4050 (100%)	9770± 3592 (100%)

Traveler Salesman Solution (DCS)

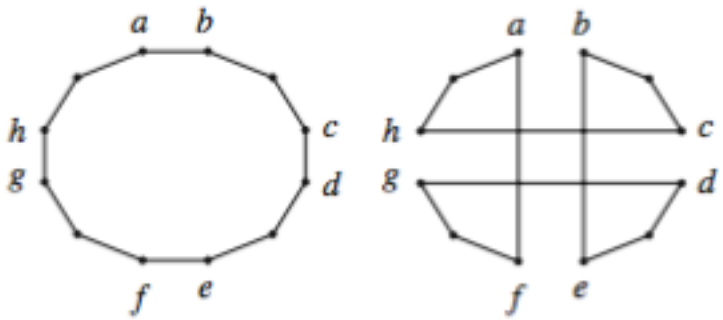
- N cities and D is distance matrix.

$$f(\pi) = \sum_{i=1}^{N-1} d_{\pi(i)\pi(i+1)} + d_{\pi(N)\pi(1)}$$

- Eggs and nests: Order of cities
- Movements



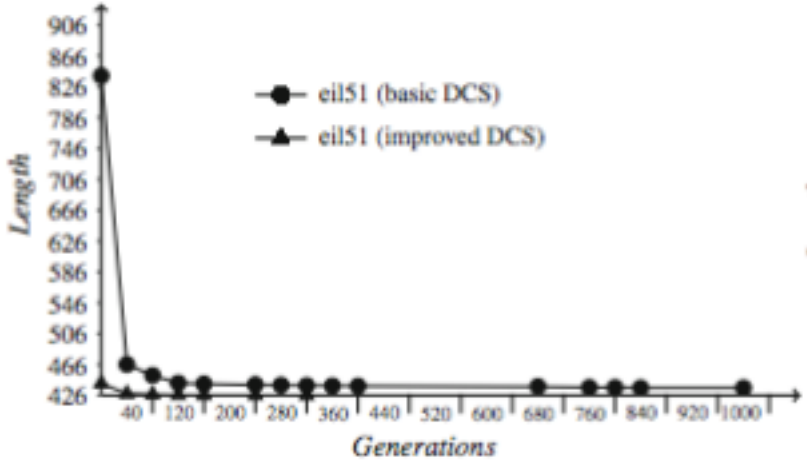
2-opt move



Double bridge move

Ouaarab et. al, (2010) *Neural Computing and Applications*, *24*(7-8), 1659-1669

Traveler Salesman Solution



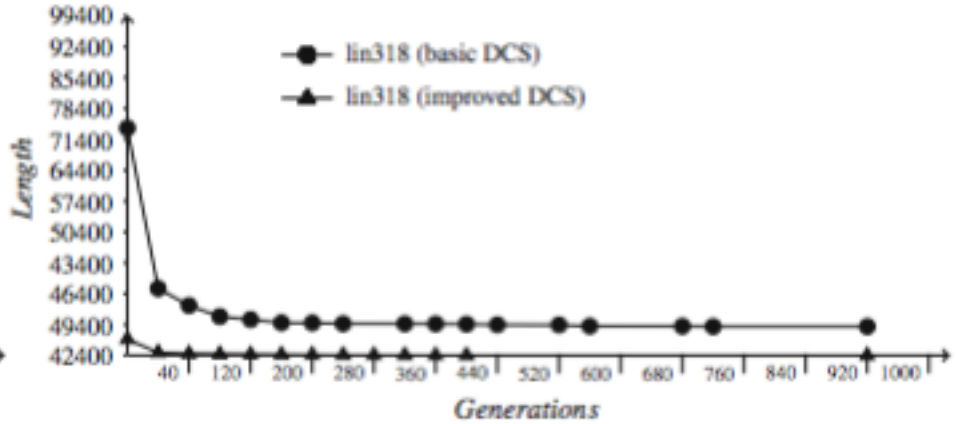


Table 4 Comparison of experimental results of the improved DCS with **GSA-ACS-PSOT** [6]

Instance	Opt	GSA-ACS-PSOT			Improved DCS		
		Best	Average	SD	Best	Average	SD
eil51	426	427	427.27	0.45	**426**	**426**	**0.00**
berlin52	7,542	**7,542**	**7,542.00**	**0.00**	**7,542**	**7,542**	**0.00**
eil76	538	**538**	540.20	2.94	**538**	538.03	0.17
kroA100	21,282	**21,282**	21,370.47	123.36	**21,282**	**21,282**	**0.00**
kroB100	22,141	**22,141**	22,282.87	183.99	**22,141**	22,141.53	2.87
kroC100	20,749	**20,749**	20,878.97	158.64	**20,749**	**20,749**	**0.00**
kroD100	21,294	21,309	21,620.47	226.60	**21,294**	21,304.33	21.79
kroE100	22,068	**22,068**	22,183.47	103.32	**22,068**	2,281.26	18.50
eil101	629	630	635.23	3.59	**629**	630.43	1.14
lin105	14,379	**14,379**	14,406.37	37.28	**14,379**	**14,379**	**0.00**
bier127	118,282	**118,282**	119,421.83	580.83	**118,282**	118,359.63	12.73
ch130	6,110	6,141	6,205.63	43.70	**6,110**	6,135.96	21.24
ch150	6,528	**6,528**	6,563.70	22.45	**6,528**	6,549.90	20.51
kroA150	26,524	**26,524**	26,899.20	133.02	**26,524**	26,569.26	56.26
kroB150	26,130	**26,130**	26,448.33	266.76	**26,130**	26,159.3	34.72
kroA200	29,368	29,383	29,738.73	356.07	29,382	29,446.66	95.68
kroB200	29,437	29,541	30,035.23	357.48	29,448	29,542.49	92.17
lin318	42,029	42,487	43,002.09	307.51	42,125	42,434.73	185.43

Ouaarab et. al, (2010) *Neural Computing and Applications*, *24*(7-8), 1659-1669

Advantages

- Simple
- Two parameters, p_a and n.
- Easy to implement.

Other use areas

- ❖Engineering optimization problems
- ❖NP-hard combinatorial optimization problems
- ❖Data fusion in wireless sensor networks
- ❖Neural network training
- ❖Manufacturing scheduling
- ❖Nurse scheduling