MODIFIED CUCKOO SEARCH: CS3

A modified Cuckoo Search.Uses rand/1/bin or local-to-best differential mutation + (in a secondary role) Lévy flights. Minimizes given objective function and prints minimum solution (objective function value and the decision variable values with which it was achieved) on screen at the end.

`runCS3opt.m`	runs the optimizer, modify as necessary for objective function at hand
`CS3.m`	optimizer code
`ofunvars.txt`	text file with decision variables and bounds (these are read by runCS3opt)

Saari J; Suikkanen H; Mendoza-Martinec C; Hyvärinen J. *Optimization of natural circulation district heating reactor primary heat exchangers.* Energies x **2023**, xxxx. doi: xxxx
Saari J; Mendoza-Martinec C; Kaikko J; Sermyagine E; Mankonen A; Vakkilainen E. *Techno-economic optimization of a district heat condenser in a small cogeneration plant with a novel greedy cuckoo search.* Energy 239 **2022**, 122622. doi:1016/j.energy.2021.122622.

---

runCS3opt

Sets up inputs for CS3 optimizer.

SYNTAX

```
runCS3opt(ofun, F_VTR)
```

DESCRIPTION

Function `runCS3opt` runs the CS3 algorithm. Current set-up is for shell-and-tube heat exchanger (STHE) optimization. The beginning part sets up the struct `S_struct` containing necessary parameters for the optimizer.

Applied to heat exchanger optimization, various economic and metal property data for objective function value calculation are included in `S_struct`.

These are passed to objective function in `FixedParams` struct

`runCS3(ofun, F_VTR)`

- `ofun` – objective function type. `'STHX2D2'` is Case 2 in Saari et al. (2019).
- `F_VTR` – Value To Reach, the objective function value that should be reached, i.e. reaching this value is a stopping criteria. Setting an intentionally unrealistically good value allows the algorithm to run until reaching maximum iterations or objective function evaluations.

CS3

Modified cuckoo search (CS3) optimizer.

Syntax

```
[FVr_bestnest,S_bestval,I_NFE] = CS3(ofun,S_struct)
```

Description

Stochastic optimizer for mixed-integer, multi-modal, multi-constrained non-linear optimization.

Based on the variant labeled “CS2” in Saari et al.(2022), originally MATLAB code in Xin-She Yang: Nature-Inspired Optimization Algorithms (2010). CS3 emphasizes differential mutation heavily over Lévy flights. When possible, input parameter nomenclature is similar to that in the Differential Evolution deopt.m file provided by Storn et al. (e.g. https://github.com/sriki18/MDEpBX-Matlab/blob/master/deopt.m)

Input variables

`ofun` – objective function name

Data Types: char | string

Example: `CS3('myFunction',S_struct)`

Optimizer will use the name to run the objective function as

```
feval(ofun,FVr_candidate,S_struct)
```

where `FVr_candidate` is a candidate solution generated by the optimizer. Functio ‘`myFunction`’ will receive as input the candidate solution and the entire `S_struct` structure.

`S_struct` – parameters for optimizer and objective function

Data Types: structure

Example: `CS3('myFunction',S_struct)`

The structure contains certain necessary fields containing control parameter values for the optimizer, as well as any number of optional constant parameters sent for the objective function upon evaluation. If any non-mandatory control parameter value is missing from the structure, a default value is used (i.e., the optimizer will still run if).

The following table lists the fields containing optimizer parameter values, and their defaults. Default value **”mand”** means the value is mandatory.

S_struct

Field	Data type	Default	Description
I_D	positive integer	**mand**	number of decision variables
FVr_minbound	floating-point vector	**mand**	decision variable lower bounds
FVr_maxbound	floating-point vector	**mand**	decision variable upper bounds
I_NP	positive integer	**mand**	Number of Parents (population size) Recommendation: 2*I_D for "easy" problems, 4…6*I_D for highly multi-modal
F_VTR	floating-point	**mand**	Value to Reach; a termination criteria. Can be set to an unachievably good value to let the optimizer to run until maximum function evaluations or iterations.
I_itermax	positive integer	**mand**	maximum number of iterations (generations); a termination criteria.
I_strategy	positive integer	1	mutation strategy for differential mutation. 1 – rand/1/bin 2 – local-to-best
I_refresh	positive integer	1	refresh cycle: output interval (for convergence plot, population print-out to file, and screen output) in generations
I_plotting	positive integer	0	toggle for collecting data for convergence plots. 0 – no convergence plot data collected 1 – matrix FM_convplot is generated to create a convergence plot in post-processing.
I_plotmax	positive integer	100	maximum index for plotting matrix (i.e. how many best-so-far objective function values are sampled throughout the run)
I_PopPrint	positive integer	0	toggle for printing out populations. 0 – no population print-out to a files 1 – writes entire population to text file CS3population_hist.txt every I_refresh iterations
I_NFEmax	positive integer	-	maximum Number of Function Evaluations; a termination criteria. I_itermax*I_NP*2 if not set otherwise (maximum theoretical maximum with I_itermax iterations and F_pa = F_pa1 = F_paMax = 1.
F_CR	floating-point	0.1	Crossover Rate: 0 ≤ CR < 1. 0 candidate solution = mutant 0.5 candidate solution half/half from mutant and base vectors 1 candidate solution = base vector (differential mutation negated completely) recommendation: 0.05 … 0.2 for non-separable problems, 0.4 … 0.5 for separable (rare)

F_alpha	floating-point	0.05	Lévy flight scaling factor: $\alpha > 0$. 0.01 little global exploration 0.1 long leaps, often bouncing from bounds
F_pa1	floating-point	0.2	used to calculate F_pa, or fraction of worst solutions where cuckoo "abandons nest" and sets out for a Lévy flight. on iteration i=1, F_pa=F_pa1 on iteration i=I_itermax, F_pa=F_paMax linear interpolation of F_pa between i=1 and i=I_itermax.
F_paMax	floating-point	0.2	
F_beta	floating-point	1.5	Lévy exponent; currently hard-coded as $\beta = 1.5$, thus has no effect
FVr_discr	integer vector	-	decision variable type: 0 – continuous 1 – discrete currently still unused in CS3 – has no effect. may be used in a possible future termination criterion based on convergence.