Economics Bulletin

Volume 33, Issue 1

Global Optimization of Some Difficult Benchmark Functions by Host-Parasite Coevolutionary Algorithm

Sudhanshu K Mishra
North-Eastern Hill University, Shillong, India

Abstract

This paper proposes a novel method of global optimization based on host-parasite co-evolution. It also develops a Fortran-77 code for the algorithm. The algorithm has been tested on 100 benchmark functions (of which the results of 32 relatively harder problems have been reported). In its search ability, the proposed method is comparable to the Differential Evolution method of global optimization. The method has been used for solving the 'completing the incomplete correlation matrix' problem encountered in financial economics. It is found that the proposed methods as well as the Differential Evolution method solves the problem, but the proposed method provides results much faster than the Differential Evolution method.

The author is immensely grateful to the anonymous referee whose elaborate and constructive comments were invaluably helpful in bringing out this paper to the present form. The remaining limitations are entirely due to the author.
Citation: Sudhanshu K Mishra, (2013) "Global Optimization of Some Difficult Benchmark Functions by Host-Parasite Coevolutionary Algorithm ", *Economics Bulletin*, Vol. 33 No. 1 pp. 1-18.
Contact: Sudhanshu K Mishra - mishrasknehu@yahoo.com.
Submitted: August 16, 2012. **Published:** January 01, 2013.

1. Introduction

Global optimization endeavors to find the optima of the functions that are non-linear, non-differentiable, non-convex or multimodal, sometimes having multiple global optimum and numerous local optima, posing insurmountable difficulties before the classical methods of optimization. One encounters such problematic functions in engineering, sciences, operations research, statistics, economics, etc. Since the mid-1950s, efforts have been made to search for a suitable method that addresses the problem of global optimization of such problematic functions.

In the pre-1975 years, the works of Box (1957), Nelder and Mead (1964) and Box (1965) were remarkable. However, the invention of the "Genetic Algorithm" by Holland (1975) ushered an era of research in global optimization. Invented in the subsequent years, the "Clustering Method" of Törn (1978), the "Simulated Annealing Method" of Kirkpatrick et al. (1983) and Cerny (1985), the "Tabu Search Method" of Glover (1986), the "Ant Colony Algorithm" of Dorigo (1992), the "Particle Swarm Method" of Kennedy and Eberhart (1995), the "Differential Evolution Method" of Storn and Price (1995), and the "Generalized Simulated Annealing method" of Tsallis and Stariolo (1995) are notable and effective methods of global optimization. Some other recently proposed methods such as the "Harmony Search" of Geem et al. (2001), the "Bee System" of Lŭcíc and Teodorovíc (2001), the "Bee Swarm Optimization" of Karaboga (2005) and Karaboga and Basturk (2007), etc. also are quite promising. Teodorović et al. (2011) is a rich source of information on "Bee Colony Optimization".

Yang and Deb (2009) proposed a new method of global optimization based on the behavior of cuckoos that are parasitic in laying their eggs in the nests of other birds (such as crows) who serve as hosts to hatch their eggs to chicks. It was shown that the so-called "Cuckoo Search" algorithm may prove to be quite effective for global optimization. Subsequent investigations made by Yang and Deb (2010), Civicioglu and Besdok (2011), Rajabioun (2011) and Valian et al. (2011) further demonstrated that the "Cuckoo Search" algorithm, in its original or improved version, may be very effective. The method has been tested on a large battery of benchmark (test) functions of varied difficulty levels.

The original "Cuckoo Search algorithm" of Yang and Deb (2009) or its variants (or its improved versions) is based on the idea of how cuckoos lay their eggs in the host nests, how, if not detected (and destroyed), the eggs are hatched by the hosts, how the cuckoo chicks later join the population of cuckoos and how a mathematical representation of all these can be used to search for the global optimum of a function. In brief, the algorithm may be conceptually summarized in the following four idealized rules (Yang and Deb, 2009):

1. Each cuckoo lays a single egg into a randomly chosen host-nest, while there are n nests;
2. The nests with better quality eggs (implying better fitness value of the optimand function), if not detected, would be hatched to be the cuckoo chicks, who would join the next generation;
3. The number of available host nests is fixed. The host can detect the alien egg with a probability [0, 1] and, if detected, it will either destroy the egg or abandon the nest so as to build a new nest elsewhere;
4. When generating new solutions ($x_i^{(t+1)}$) from the old one ($x_i^{(t)}$), Lévy flight is performed with the parameter $1 < \beta < 3$ and, thus, $x_i^{(t+1)} = x_i^{(t)} + \alpha \circ Levy(\beta)$; for, say cuckoo i;

$\alpha = O(1)$ and $\circ$ means entry-wise multiplication (or the Hadamard product operator). The Lévy flight endows a cuckoo with a capability to take a random walk which has a power law step length distribution with a heavy tail. It has been found (Brown et al., 2007; Pavlyukevich, 2007) that Lévy flights characterize an appropriate type random walk in many real life situations (Viswanathan et al., 1996; 1999; 2002).

2. The Objective of this Paper

It may be noted that the "Cuckoo Search" algorithm has nothing to say as to how the host birds will regenerate their nests in view of the parasitic intruders (cuckoos) and how the two (the cuckoos and the hosts) will co-evolve. The probability of detection also is an exogenously fixed number. Thus, the host birds are immune to the experience of invasion by the parasite cuckoos. However, Davies and Brooke (1989a; 1989b) and Lotem et al. (1995) observe that co-evolution does take place and the arms race theory (Dawkins and Krebs, 1979) would suggest that, in the long run, hosts should evolve good discrimination ability (and the probability of detection as high as 65%), forcing the cuckoos to switch to a new, non-discriminating host (Davies and Brooke, 1989b; Rothstein, 1990). In view of this, in a given area, where the cuckoos and the hosts interact, the rate of rejection would increase over time. Both the cuckoos and the hosts also change their strategies.

The objective of this paper is, therefore, to incorporate the co-evolutionary changes into the "Cuckoo Search" algorithm and test the efficiency of the two populations (of the parasites, say cuckoos and the hosts, say crows) in finding the global optimum of some benchmark functions. This new suggested algorithm may be called the "Host-Parasite Co-evolutionary" or HPC algorithm.

3. The Proposed HPC Algorithm

For simplicity, let there be a parasites population (say, cuckoos) and a hosts population (say, crows). Each individual parasite as well as individual host would be represented by a point. These points will be randomly generated and would lie in the domain of the function to be minimized. Accordingly, smaller value of the function implies better fitness. Each parasitic individual will take a random flight (Gaussian/Cauchy/Burr-xii or Lévy flight with some probability) and if its post-flight fitness is better than its pre-flight fitness (failing which it would not make an attempt to lay any egg in the host nest and thus would retain its old status), then it will randomly choose a host net that has not as yet been invaded by another parasite and where the quality of the host eggs are inferior to the parasite egg. The eggs of the parasite, however, may be detected by the host and destroyed. If not detected, however, the nestling, after being hatched in the host nest, would join the parasite population. Only the best parasites, however, will enter into the next generation. The algorithm may be outlined as follows:

1. At the start, randomly generate n_c parasite individuals ($x^{(t)}$) and n_k host individuals ($y^{(t)}$) as the points in the m dimensional domain of the function $f(.)$ to be optimized. Evaluate them for $f(x^{(t)})$ and $f(y^{(t)})$. Arrange each population (of the parasites and the hosts) such that $f(x_1^{(t)}) \le f(x_2^{(t)}) \le ... \le f(x_{nc}^{(t)})$ and $f(y_1^{(t)}) \le f(y_2^{(t)}) \le ... \le f(y_{nk}^{(t)})$. It may be noted, however,

that such an arrangement is not necessary for the algorithm to work. At this stage, the parenthesized superscript (t) takes on a value of zero ($t = 0$) that denotes initialization.

2. Let each parasite individual $x_i^{(t)}$ randomly choose a host individual $y_j^{(t)}$ and make an effort to update itself in each of its m coordinates in view of (i) a random flight, (ii) a random direction ±, and (iii) difference between its own coordinates and the chosen host's (corresponding) coordinates. That is, $x_i^{(t)} \Leftarrow x_i^{(t)} + \delta x_i^{(t)}$ if $f(x_i^{(t)} + \delta x_i^{(t)}) < f(x_i^{(t)})$, where the random flight $\delta x_i^{(t)} = a \circ r \circ \varphi(\beta) \circ (y_j^{(t)} - x_i^{(t)})$ for each coordinate of $x_i^{(t)}$. More explicitly stated, $\delta x_{ik}^{(t)} = a_k .\ r_k .\ \varphi_k(\beta).(y_{jk}^{(t)} - x_{ik}^{(t)})$ for $k = 1, 2, ..., m.$ Here r is an array of uniformly and independently distributed random numbers in $(-0.5,\ 0.5)$, a is an array of independent $(1/2)\Gamma$-distributed random numbers in $(0,1)$ and $\varphi(\beta)$ is an array of independent random numbers that effects a random flight (such as Gaussian, Cauchy, Burr-xii or Lévy). Each of the three arrays of random numbers (r, a and $\varphi(\beta)$) has m elements. The symbols $\Leftarrow$ and $\circ$ stand for 'is replaced by' and Hadamard or 'element-wise' multiplication respectively. However, if $f(x_i^{(t)} + \delta x_i^{(t)}) \geq f(x_i^{(t)})$, then $x_i^{(t)}$ maintains its status quo.
3. Let each parasite, $x_i^{(t)}$, that improved itself in step 2 above make an attempt to lay its eggs in the nest of a randomly chosen host, $y_j^{(t)}$, provided that (i) the chosen host net does not as yet contain any parasite eggs, (ii) $f(x_i^{(t)}) < f(y_j^{(t)})$ and (iii) the attempt is not foiled by the chosen host. The host population has an evolving detection function.
4. Based on the success of individual parasites in step 3 above, $p^{(t)}$, the probability of the success of the parasite population over-the-generations is updated (elaborated in the sub-section below).
5. The host individuals update themselves as in step 2 above ($y_i^{(t)} \Leftarrow y_i^{(t)} + \delta y_i^{(t)}$ if $f(y_i^{(t)} + \delta y_i^{(t)}) < f(y_i^{(t)})$), but using a flight with slightly different parameters such that the random flight $\delta y_i^{(t)} = \omega \circ \rho \circ \varphi(\gamma) \circ (x_j^{(t)} - y_i^{(t)})$. Here ρ is an array of uniformly and independently distributed random numbers in $(-0.5,\ 0.5)$, ω is an array of independent $(1/2)\Gamma$-distributed random numbers in $(0,1)$ and $\varphi(\gamma)$ is an array of independent random numbers that effects a random flight (such as Gaussian, Cauchy, Burr-xii or Lévy). Each of the three arrays of random numbers (ρ, ω and $\varphi(\gamma)$) has m elements. If $f(y_i^{(t)} + \delta y_i^{(t)}) \geq f(y_i^{(t)})$, then $y_i^{(t)}$ maintains its status quo.
6. Arrange each population (of the parasites and the hosts) such that $f(x_1^{(t)}) \leq f(x_2^{(t)}) \leq ... \leq f(x_{nc}^{(t)})$ and $f(y_1^{(t)}) \leq f(y_2^{(t)}) \leq ... \leq f(y_{nk}^{(t)})$. It may be noted that such an arrangement is not necessary.
7. The superscript (t) takes on a value incremented by 1.
8. Go to step 2 if the termination conditions are not met.

3.1. The Detection function of the Parasite Eggs by the Host

The parasites are able to survive to the next generation, first, if their eggs survive the detection by the host and, secondly, if they beget smarter offspring. The value of $p^{(t)}$, the probability of surviving the detection of their eggs laid in the host nest at generation (or iteration) t, is an over-the-generations cumulative probability given as $p^{(t)} = \sum_{g=1}^{t} n_s^{(g)} / (t \times n_c)$ while $n_s^{(g)} + n_u^{(g)} = n_c^{(g)} = n_c$ or the ratio of the total number of all successful parasite individuals over the generations ($n_s^{(g)}$; $g=1, t$) to the total number of all successful ($\sum_{g=1}^{t} n_s^{(g)}$) plus unsuccessful ($\sum_{g=1}^{t} n_u^{(g)}$) parasites over the generations, each generation having n_c individuals. Over the generations, $p^{(t)}$ decreases while $pd^{(t+1)}$ increases. In turn, $pd^{(t)}$ affects the success rate of the parasites (and therefore $p^{(t+1)}$) in the next generation making the system co-evolutionary. This is modelled on the basis of observations in the real world that suggest an 'over-the-generations' increasing incidence of detection of the parasite eggs by the host population, ultimately forcing the parasites to shift to new or different hosts who have not yet adapted themselves to the skills of the parasites.

Table-1. Comparison of Alternative Detection Functions in HPC (with Levy Flights) for Search of Optimal Values of the Weierstrass Function of Different Dimensions (D)

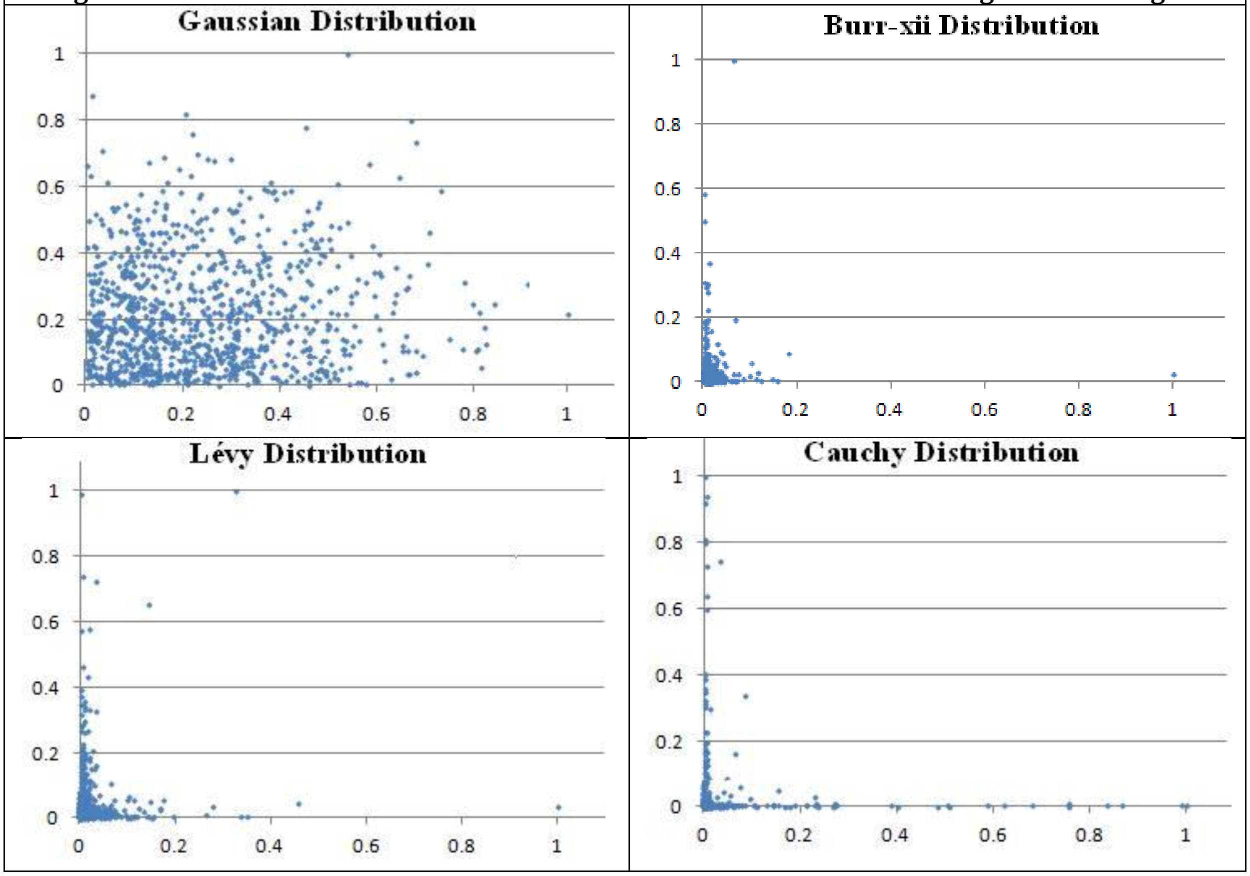
D	Gompertz			Logistic			Logit			Linear		
	Opt value	SD	T	Opt value	SD	T	Opt value	SD	T	Opt value	SD	T
10	8.63E-11	1.04E-11	1.355	8.16E-11	1.21E-11	1.338	8.24E-11	9.01E-12	1.382	8.75E-11	1.27E-11	2.283
20	9.00E-11	7.04E-12	4.607	9.21E-11	7.73E-12	4.587	1.21E-10	7.62E-11	4.771	8.98E-11	6.71E-12	4.610
30	9.50E-11	3.81E-12	10.382	9.23E-11	9.16E-12	10.294	1.89E-10	2.23E-10	10.721	9.33E-11	3.53E-12	10.323
40	9.45E-11	3.50E-12	19.327	9.55E-11	2.84E-12	19.457	1.78E-10	1.76E-10	19.568	9.48E-11	5.39E-12	19.409
50	9.55E-11	5.33E-12	32.551	9.51E-11	3.25E-12	33.082	9.55E-11	4.88E-12	33.166	1.05E-10	2.03E-11	32.772
60	9.49E-11	5.50E-12	52.172	9.65E-11	3.87E-12	51.651	9.70E-11	2.97E-12	51.894	9.61E-11	2.60E-12	51.944
70	9.58E-11	3.11E-12	78.580	9.66E-11	3.25E-12	78.279	9.71E-11	2.64E-12	76.993	9.91E-11	1.03E-11	78.509

The detection function, $pd^{(t+1)} = \psi(p^{(t)})$, of the parasite eggs by the host may be a linear function such as $pd^{(t+1)} = \alpha(1-p^{(t)})$ or a sigmoid function such as logistic function ($pd^{(t+1)} = \alpha_0 - \alpha(1+\exp(-p^{(t)}))^{-1}$), logit function ($pd^{(t+1)} = -\alpha \ln(p^{(t)}/(1-p^{(t)}))$) or Gompertz function ($pd^{(t+1)} = \alpha \exp[-2\exp\{-(1+\ln(1+p^{(t)}))^{-1}\}]$) with the scaling factor α in $(0,1)$ and the constant α_0 in $(0,1)$. As it has been reported in Table-1, there is no clear advantage in using the one function over the other, although there is a weak indication that the Gompertz detection function may provide more consistent results.

3.2. Choice of the Random Flight-Generating Function

Among many possible choices that may be made regarding the flights taken by the parasites (and the host), Burr-xii, Cauchy, Gaussian or Lévy distribution (or any one among many others) may be worth consideration for specifying $\varphi(.)$. A graphical presentation of 1000 points in 2-dimensional space for these distributions is given in Fig.-1. Of these, Burr-xii, Cauchy and Lévy distributions are heavy-tailed. It has been found (Viswanathan et al. 1999; Gutowski, 2001; Pavlyukevich, 2007; Yang and Deb, 2009) that the random flights generated by heavy-tailed distributions, especially the Lévy distribution, perform better in escaping a convergence to local optima and also are frequently observed in the animal behavior. This has been corroborated by optimizing some select test functions (Table-2) where Lévy flights obtain best values most consistently, followed by Cauchy flights (that obtain the worst value only once) and Burr-xii flights. Gaussian flights perform worst. These findings suggest that Lévy and Cauchy flights may be used alone or in conjunction, with lager (say 95%) and smaller (5%) probabilities, respectively.

Fig.-1. Scatter of 1000 Points in 2-D Under Different Distributions Effecting Random Flights



.

Table.-2. Performance of Some Select Benchmark Functions with Different Types of Random Flight

Function (Dimension)	Gaussian Flight		Burr-xii Flight		Lévy Flight		Cauchy Flight	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Giunts (2)	0.0644704444	9.313E-10	0.0644704444	0	0.0644704444	0	0.0644704444	0
Easom (2)	-1	0	-1	0	-1	0	-1	0
Trefethen (2)	-3.30686865	7.300E-08	-3.30686865	7.300E-08	-3.30686865	7.300E-08	-3.30686865	7.30E-08
Levy-8 (3)	6.737666E-13	2.528E-13	*7.24103E-13*	1.919E-13	6.531185E-13	2.637E-13	**5.001159E-13**	2.40E-13
Perm-1 (4)	*1.506391E-07*	4.34E-07	**1.570331E-09**	2.232E-09	5.054093E-09	9.077E-09	3.688489E-08	1.22E-07
Shekel (4)	*-8.4538657*	2.403221	-8.79373249	2.2544213	-9.81333288	1.271665	**-10.1531997**	2.06E-07
Hougen (5)	*0.331174304*	0.0070644	0.319942193	0.0110459	0.323038786	0.0068002	**0.314789813**	0.008429
Powel (8)	4.535538E-07	1.477E-07	*5.100929E-07*	1.506E-07	**3.673879E-07**	1.780E-07	4.81744E-07	1.89E-07
ANNS-XOR (9)	0.959758757	1.054E-08	*0.959758757*	1.490E-08	0.959758757	0	**0.95975899**	5.96E-07
Schwefel (10)	*-2003.66884*	804.4180	**-4189.82887**	0.000136	**-4189.82887**	0.000136	**-4189.82887**	0.000136
Quintic (10)	**1.651056E-08**	3.904E-08	*4.43242E-06*	1.378E-05	1.550833E-07	5.799E-07	1.742966E-06	4.79E-06
Ackley (**1**0)	9.178140E-13	9.178E-13	9.123665E-13	7.843E-14	**8.974451E-13**	8.383E-14	*9.441041E-13*	4.75E-14
Michalewicz (10)	*-9.59941626*	0.0363532	-9.6594969	0.001669	**-9.66015172**	1.686E-07	**-9.66015172**	2.06E-07
Rosenbrock (10)	*0.0131510732*	0.0166608	6.577238E-07	4.440E-07	1.847974E-07	3.806E-07	**9.488363E-13**	4.21E-14
Trigonomet (10)	*2.208646E-05*	2.471E-05	2.654573E-06	8.816E-06	1.015045E-07	2.695E-07	**2.236183E-10**	7.41E-10
Lunacek (10)	*27845.7836*	65956.24	12258.8264	15213.408	**8.569804E-13**	9.285E-13	10269.5682	14671.83
Weierstrass (20)	*9.431271E-13*	4.15E-14	**9.066525E-13**	6.578E-14	9.412323E-13	5.899E-14	9.369690E-13	3.87E-14
Keane-Bump (60)	*-0.805173227*	0.009944	-0.829840357	**0.0037952**	**-0.836701304**	0.001166	-0.836449067	0.001199

Note: Results based on 15 trials. Max run time and no. of agents in search are presented in Table-4. The worst values are in the italics and the best values are in bold. Mean is the arithmetic average and SD stands for standard deviation. (accuracy level = 1.0e-12).

3.3. Complexity and Mean Time of Successful Search by the HPC Algorithm

In Table-3 we present the mean search time and the associated statistics of the proposed HPC algorithm and compare them with the DE search algorithm. The Weierstrass function, well known for being 'everywhere continuous but nowhere differentiable' and having a theoretically known min value = 0 in [0,1], has been used as a test case. The success of the two algorithms is adjudged if the search outcome is very close to the minimum value (less than 1.0e-10). For HPC we have used 60 agents ($n_c = 30;\ n_k = 30$ for all dimensions) and Lévy flight alone. For DE we have used dimension-dependent number of agents, since it is suggested that the number of agents must not be less than four times the dimension of the function to be optimized, although, very often, this factor is 10, rather than the minimal four. We have used this factor = 10 (Brest et al., 2006). It may be reported in passing that for dimension = 60, the DE often fails when the number of agents is merely 4D =240. Except for the dimensions 40, 50, 60 and 70, for which the DE works for 5, 5, 3 and 3 replicates respectively (since it demands a long time for 15 replications), both the algorithms have been used to obtain 15 replicates and mean-time and mean-near-optimum values are reported for the same.

For the HPC algorithm we get $T = 9.369546 - 0.8418D + 0.026538D^2;\ R^2 = 0.9963$ and for the DE we get $T = 210.513 - 21.3076D + 0.5143D^2;\ R^2 = 0.9802$ as relationships between dimension (D) and the mean time (T) required for successfully searching the near-optimal value of the Weierstrass function. It may be noted that while the DE suffers from the 'curse' of dimensionality (as it requires the number of agents to increase with dimension), the HPC does not have such limitation. All these computations (as well as elsewhere in this paper) are carried out on an HP Desktop Personal Computer with 2.4 GHz Core-2 Duo CPU. The computation time (T) is the CPU time in seconds (obtained by calling the internal function CPU_TIME(.) in FORTRAN 77 providing computation time up to microseconds).

Table-3. Mean CPU Time of Execution for Search of Optimal Values of the Weierstrass Function of Different Dimensions by HPC and DE Algorithms

Sl No.	Dim	T = Mean(time)		SD(time)		Mean(opt-value)		SD(opt-value)	
		HPC	DE	HPC	DE	HPC	DE	HPC	DE
1	10	1.3552	5.45521	0.0315	0.01451	8.62563E-11	0	1.04434E-11	0
2	20	4.6073	20.6792	0.0857	0.07326	9.00149E-11	2.43484E-11	7.03629E-12	1.35562E-11
3	30	10.3823	91.31875	0.1315	0.04067	9.49669E-11	0	3.80977E-12	0
4	40	19.3271	200.91(5)	0.2726	0.06203	9.45344E-11	4.26326E-14	3.50301E-12	2.54211E-14
5	50	32.5510	374.416 (5)	0.4144	0.21392	9.54780E-11	5.68434E-13	5.32514E-12	3.28144E-13
6	60	52.1719	716.677 (3)	0.5352	127.51	9.49276E-11	3.76562E-05	5.49780E-12	5.32539E-05
7	70	78.5802	1298.81(3)	0.6424	230.66	9.58096E-11	0.00276423	3.11242E-12	0.003904219
8	80	114.318	NO	1.3412	NO	9.61942E-11	NO	4.30669E-12	NO

Notes: (i) CPU time in seconds, (ii) Opt-value = nearest optimal value for 1.0e-10 accuracy requirement (except DE for D=60 and 70), (iii) Mean time for DE for Dim = 40 and 50 is for 5 replicates; for Dim = 60 and 70 for 3 replicates. Elsewhere Mean time is for 15 replicates. NO = Not Computed due to large run-time requirement. Only Lévy flights are used for HPC. .

4. Performance of the HPC Algorithm on Some Benchmark Functions

To test the effectiveness of the proposed HPC algorithm we have used 32 benchmark functions (Table-4). Many of the selected benchmark functions are quite difficult to optimize (Mishra, 2010; 2006a; 2006b). Some functions, namely Lunacek (Dieterich and Hartke, 2012), modified

Lunacek, Eggholder, Keane, ANNS XOR, Easom, Perm-1, Perm-2, Hougen, AMGM, Michalewicz, etc are quite hard to optimize.

For each benchmark function, the proposed HPC and the Differential Evolution (DE) algorithm are run 15 times with different random number seeds and for varying time limits for the search run to terminate. In all cases, the population of agents in the HPC is constant (30 for host and 30 for parasite). In case of DE, however, the population of agents is 10 times the dimension of the function concerned.

4.1. Relative Performance of the HPC

Relative performance of an optimization algorithm that yields its best value (v_1) with respect to the best value known (v_0) and the best value obtained by another competing algorithm (v_2) may ordinarily be measured as $(v_1 - v_0)/(v_2 - v_0)$. However, when v_0 and v_2 are zero (and more so when v_1 also is zero), this measure may not be computable. Furthermore, computationally, such a measure of performance may be unstable (and perhaps misleading) when v_1 or v_2 is near-zero, viz. $abs(v_1) < 1.0E-10$; $abs(v_2) < 1.0E-10$. Therefore, we propose that, first, near-zero values may be considered as zero and, secondly, the relative measure of performance (c) be obtained as $c = \exp(b-a)$, where $a = abs(v_1 - v_0)$ and $b = abs(v_2 - v_0)$. This measure, c, would take on a value of unity if the proposed and the competing algorithms are performing equally well and less than (greater than) unity if the former performs worse (better) than the latter. Accordingly, in Table-4, we find that for Easom, Power-Sum, Hougen, Perm-2 and Kene-Bump, the HPC performs better than the DE. For Shekel, Quintic, Rosenbrock, Lunacek and Mod-Lunacek, the HPC performs worse than the DE and, in particular for the Eggholder function, the performance of HPC is dismal. For the rest (21 benchmark functions), HPC and DE perform equally well.

Table-4. Global Optimum of Select Benchmark Function Obtained by Differential Evolution (DE) and Host-Parasite Co-Evolution (HPC) Algorithms

Function (Dimension)	Best Value Known	Mean Value (15 runs)	Standard Deviation (15 runs)	Efficiency of HPC (c)	No. of Agents	CPU Time (in second)
Giunta(2)	0.0644704444	0.0644704444	0	1	(30, 30)	2/2
		0.0644704444	9.31322575E-10	1	(30, 30)	2/1.8667
		0.0644704444	0	-	20	2/0.0010
Easom(2)	-1	-1	1.82501207E-08	2.718276	(30, 30)	2/2
		-1	1.82501207E-08	2.718276	(30, 30)	2/0.0073
		2.16055169E-06	4.46970101E-06	-	20	5/0.0615
Trefethen(2)	-3.306869	-3.30686865	7.3000483E-08	1	(30, 30)	2/2
		-3.30686865	7.3000483E-08	1	(30, 30)	2/2
		-3.30686865	7.3000483E-08	-	20	2/ 0.0395
Shubert(2)	-186.730909	-186.730909	2.6973983E-06	1	(30, 30)	2/2
		-186.730909	2.6973983E-06	1	(30, 30)	2/2
		-186.730909	3.81469727E-06	-	20	2/ 0.0406
Levy-8(3)	0	4.8663378E-13	2.56378772E-13	1	(30,30)	1/0.0094
		4.68290849E-13	2.90351267E-13	1	(30,30)	1/0.0094
		1.49966072E-32	1.49966072E-32	-	30	2/0.0177
Hartmann(3)	-3.86277761	-3.86277761	1.1151008E-07	1	(30,30)	2/2
		-3.86277761	1.1151008E-07	1	(30,30)	2/2
		-3.86277761	1.1151008E-07	-	30	2/ 0.0563
Perm-1(4)	0	5.15007117E-09	8.98582283E-09	1	(30,30)	60/60
		4.57128986E-08	9.4498728E-08	1	(30,30)	60/60

		1.91395519E-14	6.0483572E-14	-	40	60/0.1313
Power-Sum(4)	0	2.18300406E-07	3.86356388E-07	1.009052	(30,30)	10/9.7521
		4.71195971E-07	1.01879405E-006	1.009051	(30,30)	10/10
		0.00901106704	0.010173588	-	40	10/ 0.1281
Wood(4)	1.094854	1.09485393	1.49011612E-008	1	(30,30)	2/2
		1.09485393	1.49011612E-08	1	(30,30)	2/2
		1.09485393	2.10734243E-08	-	40	2/0.0469
Shekel(4)	-10.4832696	-9.81333288	1.27166511	0.511741	(30,30)	10/10
		-9.47346609	1.73298743	0.36429	(30,30)	10/10
		-10.4832696	1.68587394E-07	-	40	10/0.05625
Colville(4)	0	1.00374576E-12	2.13129483E-14	1	(30,30)	5/3.8104
		1.23403554E-12	6.76298926E-13	1	(30,30)	5/3.7323
		0	0	-	40	5/ 0.0698
Hougen(5)	0.298901	0.32307247	0.0119201832	1.010182	(30,30)	10/10
		0.315743327	0.0133511609	1.017613	(30,30)	10/10
		0.333202566	0.0568826927	-	50	10/ 0.2813
Glankwahmdee(5)	-739.822991	-739.822991	0	1	(30,30)	2/2
		-739.822991	1.07895932E-05	1	(30,30)	2/2
		-739.822991	1.52587891E-05	-	50	2/0.1677
Powel(8)	0	3.32358613E-07	1.89058359E-07	1	(30,30)	10/10
		4.56407089E-07	1.42100592E-07	1	(30,30)	10/10
		1.82599811E-15	3.78464179E-15	-	80	10/ 0.0917
ANNS-XOR (9)	0.959759	0.959758757	1.82501207E-08	1	(30,30)	3/3
		0.959758757	1.82501207E-08	1	(30,30)	3/3
		0.959758757	1.49011612E-08	-	90	10/ 2.3938
Quintic(10)	0	3.81976838E-06	1.42583882E-05	0.999996	(30,30)	10/4.1917
		7.99360578E-13	1.77635684E-13	1	(30,30)	10/4.0521
		0	0	-	100	10/ 0.1802
Perm-2(10)	0	0.00484731748	0.0180073135	1.024009	(30,30)	10/10
		4.48718087E-05	3.1249553E-05	1.028939	(30,30)	10/10
		0.0285727536	0.023623502	-	100	10/4.9031
AMGM(10)	0	7.33396949E-10	2.86802073E-010	1	(30,30)	10/10
		6.85353416E-10	3.31775455E-10	1	(30,30)	10/10
		5.12102751E-14	1.01167967E-13	-	100	10/0.1458
Griewank(10)	0	8.62850532E-13	1.14947296E-13	1	(30,30)	1/0.1396
		7.83151322E-13	1.8979903E-013	1	(30,30)	1/0.1479
		4.69624339E-14	9.08976143E-14	-	100	2/0.3385
Rastrigin(10)	0	7.97110526E-13	1.48411705E-13	1	(30,30)	1/0.3719
		8.47677484E-13	1.52556134E-13	1	(30,30)	1/0.3938
		0	0	-	100	2/ 0.2365
Ackley(10)	0	8.92234434E-13	9.18815984E-14	1	(30,30)	2/0.2042
		9.23498315E-13	5.69538405E-14	1	(30,30)	2/0.2104
		2.07241631E-16	8.86202492E-16	-	100	2/0.3281
Michalewicz(10)	-9.66015172	-9.66015172	2.06476546E-007	1	(30,30)	2/2
		-9.66015172	2.06476546E-07	1	(30,30)	2/2
		-9.66015172	1.1920929E-07	-	100	2/0.6177
Schwefel(10)	-4189.829	-4189.82887	0.000136478758	1	(30,30)	3/3
		-4189.82887	0.000136478758	1	(30,30)	3/3
		-4189.82887	0.000136478758	-	100	3/0.3115
Paviani(10)	-45.77848	-45.7784755	0	1	(30,30)	2/2
		-45.7784755	1.16800773E-06	1	(30,30)	2/2
		-45.7784755	0	-	100	2/0.3302
Rosenbrock(10)	0	8.28664889E-07	1.72367926E-06	0.999999	(30,30)	10/10
		1.77119335E-07	3.82418968E-07	1	(30,30)	10/10
		4.59534037E-14	1.41867338E-13	-	100	10/0.2333
Trigonometric(10)	0	6.23321862E-10	2.29366091E-09	1	(30,30)	2/0.9698
		6.32433549E-12	1.42523336E-11	1	(30,30)	2/0.6948
		2.01553787E-14	7.48937185E-14	-	100	2/ 0.4354
Lunacek(10)	0	0.114383586	0.427984188	0.891916	(30,30)	3/0.8448
		7.99827695E-13	1.43589822E-13	1	(30,30)	3/0.8448
		1.23819177E-13	2.51442617E-13	-	100	3/0.2844
Mod-Lunacek(15)	0	20.0436728	2.45420149	5.58E-09	(30,30)	5/5
		20.3890035	4.72483206	3.95E-09	(30,30)	5/5

		20.3890035	4.72483206	-	150	5/3.2896
Eggholder(15)	-12875.5766	-12111.6066	513.576372	0	(30,30)	60/60
		-12351.9542	340.858877	0	(30,30)	60/60
		-12472.523	240.801769	-	150	60/ 5.6792
Trid(20)	-1520	-1520	0	1	(30,30)	12/12
		-1520	0	1	(30,30)	12/12
		-1520.	2.15791864E-05	-	200	12/1.0583
Weierstrass(20)	0	8.47914331E-13	9.864922E-14	1	(30,30)	10/5.0854
		9.22284471E-13	8.95095853E-14	1	(30,30)	10/5.2823
		0	0	-	200	35/31.0510
Keane-Bump(60)	Not known	-0.836201	0.00130014145	1.146462	(30,30)	60/60
		-0.837407863	0.000787192094	1.147846	(30,30)	60/60
		-0.69952056	0.0118118307	-	600	130/73.367
For every benchmark function the first two rows relate to HPC with mixed flights. Row-1 relates to Lévy flights (with probability = 95%) and Cauchy flights (with probability = 5%) and row-2 relates to Lévy flights (with probability = 95%) and exp(Cauchy) flights (with probability = 5%). The third row relates to DE. For HPC agent (nc, nk) =(parasite, host) populations and for DE the no. agents is 10 · D. As to run time, T refers to CPU time allowed/time taken. The 15 random number seeds used are: 45331, 44431, 44421, 44401, 45671, 53277, 34567, 23171, 98267, 49821, 11387, 17869, 12352, 12017 and 10501. Minimum accuracy for termination is 1.0E-12.						

4.2. A Constrained Optimization Case when the HPC Clearly Outperforms the DE

In particular, a mention may be made of the Keane's Bump function, which makes a well known nonlinear constrained optimization problem hard to optimize. Its optimal values for different dimensions are not known. Mishra (2007a) obtained the value of min(Keane(60)) = -0.835835669 by the DE algorithm and min(Keane(60)) = -0.837746743 by the Repulsive Particle Swarm algorithm. The HPC algorithm obtains min(Keane(60)) = -0.838309996 (using parasite and host populations of 50 each and allowing for 200 seconds of run), which is better than both (but not yet optimal). The coordinates of the min(Keane(60)) are given in Table-5. In this regard, therefore, the performance of the HPC algorithm is remarkable.

Table-5. Coordinates of the Keane's Bump Function (Dimension = 60) obtained by HPC Algorithm

6.292849	6.245672	3.176372	3.139257	3.135303	3.138552	3.116267	3.103339	3.088621	3.082445
3.076263	3.063221	3.043688	3.037078	3.026641	3.019054	2.992814	3.002117	2.982061	2.981301
2.934628	2.945212	2.945505	2.926539	0.485694	0.482363	0.475488	0.483270	0.472872	0.481436
0.456696	0.478872	0.489860	0.468501	0.472842	0.472607	0.445679	0.463993	0.458094	0.441825
0.447224	0.435700	0.449929	0.454840	0.460273	0.452506	0.451221	0.463613	0.442843	0.430523
0.451609	0.417102	0.441924	0.432519	0.431536	0.435202	0.421458	0.425350	0.442531	0.436222

5. An Application to Financial Economics

In financial economics, correlation matrices are very important objects of study that make one of the cornerstones of Markowitz's theory of optimal portfolios (Laloux et al., 1999). Their relevance in financial analysis is demonstrated in Chesney and Scott (1989), Heston (1993), Schöbel and Zhu (1999), Xu and Evers (2003), Andersen et al. (2006), Münnix et al. (2012), etc. Correlation matrices are also used to forecast demand for a group of products (Tyagi and Das, 1999). There are three oft-studied issues with regard to the correlation matrices of financial variables: their relationship with the random matrix theory (Markowitz, 1952; Ormerod and Mounfield, 2000; Bouchaud and Potters, 2003; Potters et al., 2005), the best resolution of an invalid (non-positive semi-definite) correlation matrix into a valid (positive semi-definite) correlation matrix (Rebonato and Jäckel,1999; Higham, 2002; Anjos et al., 2003; Pietersz and Groenen, 2004; Grubisic and Pietersz, 2004; Mishra, 2004, etc.) and completion of an incomplete correlation matrix having some elements missing (Grone et al., 1984; Barett et al., 1989; Helton et al., 1989; Johnson, 1990; Barett et al., 1998; Laurent, 2001; Kahl and Günther, 2005, Mishra, 2007b, etc.).

The problem of completing the (incomplete) correlation matrix admits non-unique solution, and, therefore, some authors have suggested numerical methods that provide ranges to different unknown elements of the incomplete correlation matrix. Stanley and Wang (1969), Glass and Collins (1970), Olkin (1981) and Budden et al. (2007) have suggested very efficient methods to find such ranges for the unknown elements of very small correlation matrices (of order 4 or less). Candés and Recht (2008) approaches the completion problem by convex optimization. Mishra (2007b) provides a routine based on a stochastic search through the Differential Evolution method of global optimization.

Semi-definite completion of an incomplete correlation matrix (Nagy et al. 2012) may yield a result matrix with zero determinant implying that at least one of the financial variables generating such a matrix is entirely redundant. This is clearly unrealistic. Therefore, in this exercise we choose the problem of completing an incomplete matrix by the elements that maximize the determinant of the resulting full and valid (positive definite) correlation matrix. Johnson (1990) has pointed out that a determinant-maximizing positive-definite completion result of an incomplete (correlation) matrix is unique. Furthermore, if the vacant cells (occupied by the unknown r_{ij}, call them $\hat{r}_{ij}$), are filled by the determinant-maximizing $\hat{r}_{ij}$ then the corresponding cell(s) of $S = \hat{R}^{-1}$ would have $s_{ij} = 0$. We exploit this property in our determinant-maximizing positive-definite matrix-completion exercise.

Table-6. Correlation Matrix (R) of Daily Stock Returns of 10 Companies Traded at the New York Stock Exchange During 2001-2003.

	AIG	IBM	BAC	AXP	MER	TXN	SLB	MOT	RD	OXY
AIG	1.000	0.413	0.518	0.543	0.529	0.341	0.271	0.231	0.412	0.294
IBM	0.413	1.000	0.471	0.537	0.617	0.552	0.298	0.475	0.373	0.270
BAC	0.518	0.471	1.000	0.547	0.592	0.400	0.258	0.349	0.370	0.276
AXP	0.543	0.537	0.547	1.000	0.664	0.422	0.347	0.351	0.414	0.269
MER	0.529	0.617	0.592	0.664	1.000	0.533	0.344	0.462	0.440	0.318
TXN	0.341	0.552	0.400	0.422	0.533	1.000	0.305	0.582	0.355	0.245
SLB	0.271	0.298	0.258	0.347	0.344	0.305	1.000	0.193	0.533	0.592
MOT	0.231	0.475	0.349	0.351	0.462	0.582	0.193	1.000	0.258	0.166
RD	0.412	0.373	0.370	0.414	0.440	0.355	0.533	0.258	1.000	0.591
OXY	0.294	0.270	0.276	0.269	0.318	0.245	0.592	0.166	0.591	1.000

Source: Tumminello, M. et al. (2010), p. 42 (For details see ttp://arxiv.org/pdf/0809.4615v1.pdf)

Tumminello et al. (2010) provide the correlation matrix (reproduced in Table-6) of daily stock returns of 10 companies [namely American Intl Group Inc.(AIG), Intl Business Machines (IBM), Bank Of America (BAC), American Express Co.(AXP), Merrill Lynch (MER), Texas Instruments (TXN), Schlumberger (SLB), Motorola (MOT), Royal Dutch Pet New (RD) and Occidental Petroleum (OXY)] traded at the New York Stock Exchange during January 2001 through December 2003 (748 records), in order of their market capitalization in December 2003.

Of these, three stocks (OXY, RD, SLB) belong to the energy sector, three (IBM, MOT, TXN) to the technology sector and four (AIG, AXP, BAC, MER) to the financial sector. The determinant of this correlation matrix [det(R)] is 0.0126676649.

.

Table-7. Recovery of Missing Elements in Completing the Incomplete Correlation Matrix Problem

Expt	No. and Identification of elements obliterated/recovered	True Values (r_{ij})	Recovered Values($\hat{r}_{ij}$)	Determinant (HPC)	Determinant (DE)	Mean CPU Time (sec)
1	1 ($\hat{r}_{12}$) and ($\hat{r}_{21}$)	0.413	0.392534767	0.0126851733 (0)	0.0126851733 (0)	15 [HPC] 9.38333 [DE]
2	2 ($\hat{r}_{23}, \hat{r}_{24}$) and ($\hat{r}_{32}, \hat{r}_{42}$)	0.471 0.537	0.42034691 0.45737954	0.0130451433 (4.0327E-10)	0.0130451433 (4.0327E-10)	15 [HPC] 18.36875 [DE]
3	3 ($\hat{r}_{39}, \hat{r}_{3,10}, \hat{r}_{4,10}$) and ($\hat{r}_{93}, \hat{r}_{10,3}, \hat{r}_{10,4}$)	0.370 0.276 0.269	0.337466825 0.243398150 0.308257729	0.0128140987 (0)	0.0128140987 (2.8516E-10)	15 [HPC] 28.09375 [DE]
4	4 ($\hat{r}_{39}, \hat{r}_{3,10}, \hat{r}_{45}, \hat{r}_{4,10}$) and ($\hat{r}_{93}, \hat{r}_{10,3}, \hat{r}_{54}, \hat{r}_{10,4}$)	0.370 0.276 0.664 0.269	0.343885015 0.245786265 0.514544702 0.306210076	0.0142362378 (0)	0.0142362378 (3.2927E-10)	16 [HPC] 57.3375 [DE]
5	4 ($\hat{r}_{39}, \hat{r}_{3,10}, \hat{r}_{45}, \hat{r}_{4,10}, \hat{r}_{9,10}$) and ($\hat{r}_{93}, \hat{r}_{10,3}, \hat{r}_{54}, \hat{r}_{10,4}, \hat{r}_{10,9}$)	0.370 0.276 0.664 0.269 0.591	0.343885015 0.241710106 0.514544702 0.289174414 0.365359770	0.016442216 (2.3283E-10)	0.016442216 (4.0327E-10)	16 [HPC] 73.45625 [DE]

Note: Reported CPU time and determinants are arithmetic mean of 15 trials with the standard deviation in the parentheses.

We obliterate the elements of this matrix (r_{ij} and $r_{ji} \in R$) progressively (but arbitrarily) and use HPC and DE algorithms to recover them (as if we did not know them) – amounting to an exercise in completing the incomplete correlation matrix. Our findings are presented in Table-7. HPC as well as DE maximizes determinants although DE takes much more time than HPC.

The determinant-maximizing (positive-definite) completion of (an incomplete) correlation matrix is introducing some sort of linear independence between the financial variables i and j that is described by the missing $\hat{r}_{ij}$ and is filled by $\hat{r}_{ij}$ making the determinant-maximizing $\hat{R}$ matrix. This is opposite to the possibly determinant-zeroing exercise (making the matrix a positive semi-definite) that may yield a result matrix with zero determinant implying redundancy of some financial variables. In view of the random matrix theory (Potters et al., 2005), the determinant-maximizing solution is preferable to its counterpart.

6. Concluding Remarks

The HPC is a co-evolutionary algorithm. It becomes co-evolutionary on two accounts. First that both – the parasites and the hosts take random flights in view of themselves and their randomly selected cohort at each iteration/generation ($y_{\kappa j}^{(t)}$ and $x_{\ell j}^{(t)}$) affecting and being affected by the cohort population. This is mainly competitive. Secondly, at every subsequent iteration, the egg detection (rejection) function of the hosts, ($pd^{(t+1)}$), that depends on the cumulative success of the parasites, ($p^{(t)}$), increases and, in turn, affects $p^{(t+1)}$. This is co-evolutionary in nature. The details are available in the Fortran codes (downloadable from http://nehu-economics.info/computer-programs/cuckoo_host.txt).

It appears that the performance of the HPC algorithm is comparable to the DE algorithm, which is perhaps the most efficient algorithm for global optimization (of continuous valued non-convex functions). In particular, the HPC algorithm does not suffer from the 'curse of dimensionality' while the DE does. Yet, the HPC algorithm requires further probe into its behavior. Investigations are required in selecting appropriate detection function, type of random flight and a more effective strategy to enhance convergence.

In the HPC algorithm both the hosts and the parasites make attempts to optimize. In case of many functions (whose results have not been presented here, but they are available in the Fortran code mentioned earlier), the hosts perform better than the parasites, and in case of some other functions both perform equally well. In fewer cases, however, the parasites perform better than the hosts. It is not yet understood as to the reason behind such occurrences. It requires further investigation.

References

Andersen, T.G., T. Bollerslev, P.F. Christoffersen and F.X. Diebold (2006) "Practical Volatility and Correlation Modeling for Financial Market Risk Management" in *Risks of Financial Institutions* by M. Carey and R. Stultz, Eds., University of Chicago Press for NBER, 513-548.

Anjos, M.F., N.J. Higham, P.L. Takouda and H. Wolkowicz (2003) "A Semidefinite Programming Approach for the Nearest Correlation Matrix Problem" *Preliminary Research Report*, Dept. of Combinatorics & Optimization, Waterloo: Ontario.

Barett, W.W., C.R. Johnson and M. Lundquist (1989) "Determinantal Formulae for Matrix Completions Associated with Chordal Graphs" *Linear Algebra and its Applications* 121, 265-289.

Barrett, W.W., C.R. Johnson and R. Loewy (1998) "Critical Graphs for the Positive Definite Completion Problem" *SIAM Journal of Matrix Analysis and Applications* 20, 117-130.

Bouchaud, J.P. and M. Potters (2003) *Theory of Financial Risk. From Statistical Physics to Risk Management*, Cambridge University Press: Cambridge.

Box, G.E.P. (1957) "Evolutionary Operation: A Method for Increasing Industrial Productivity" *Applied Statistics* 6, 81-101.

Box, M.J. (1965) "A New Method of Constrained Optimization and a Comparison with Other Methods" *Comp. Journal* 8, 42-52.

Brest, J., S. Greiner, B. Bŏsković, M. Mernik and V. Žumer (2006) "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems" *IEEE Transactions on Evolutionary Computation* 10(6), 646-657.

Brown, C., L.S. Liebovitch and R. Glendon (2007) "Lévy flights in Dobe Ju/'hoansi foraging patterns" *Human Ecol*. 35, 129-138.

Cerny, V. (1985) "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm" *J. Opt. Theory Appl*. 45(1), 41-51.

Civicioglu, P. and E. Besdok (2011) "A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms" *Artificial Intelligence Review*, DOI 10.1007/s10462-011-9276-0.

Budden, M., P. Hadavas, L. Hoffman, and C. Pretz (2007) "Generating Valid 4 x 4 Correlation Matrices" *Applied Mathematics E-Notes* 7, 53-59.

Candés, E.J. and B. Recht (2008) "Exact Matrix Completion via Convex Optimization", http://www-stat.stanford.edu/~candes/papers/MatrixCompletion.pdf.

Chesney, M. and L. Scott (1989). "Pricing European Currency Options: A Comparison of the Modified Black-Scholes Model and a Random Variance Model" *Journal of Financial and Quantitative Analysis* 24, 267–284.

Davies, N.B. and M. de L. Brooke (1989a) "An experimental study of co-evolution between the cuckoo, Cuculus canorus, and its hosts. I. Host egg discrimination" *J. Anim. Ecol*. 58, 207-224.

Davies, N.B. and M. de L. Brooke (1989b) "An experimental study of co-evolution between the cuckoo, Cuculus canorus, and its hosts. II. Host egg markings, chick discrimination and general discussion" *J. Anim. Ecol.* 58, 225-236.

Dawkins, R. and J.R. Krebs (1979) "Arms races between and within species" *Proc. R. Soc. Lond. Ser. B*. 205, 489–511.

Dieterich, J.M. and B. Hartke (2012) "Empirical review of standard benchmark functions using evolutionary global optimization" http://arxiv.org/pdf/1207.4318.pdf

Dorigo, M. (1992) *Optimization, Learning and Natural Algorithms*, PhD thesis, Politecnico di Milano: Italie.

Eberhart R.C. and J. Kennedy (1995) "A New Optimizer using Particle Swarm Theory" in *Proceedings Sixth Symposium on Micro Machine and Human Science* IEEE Service Center, Piscataway: NJ, 39–43.

Geem, Z.W., J.H. Kim and G.V. Loganathan (2001) “A New Heuristic Optimization Algorithm: Harmony Search” *Simulation* 76(2), 60-68.

Glass, G. and J. Collins (1970) “Geometric Proof of the Restriction on the Possible Values of rxy when rxz and ryx are Fixed” *Educational and Psychological Measurement* 30, 37-39.

Glover F. (1986) “Future Paths for Integer Programming and Links to Artificial Intelligence” *Computers and Operations Research* 5, 533-549.

Grone, R, C.R. Johnson, E.M. Sá, and H. Wolkowicz (1984) “Positive Definite Completions of Partial Hermitian Matrices” *Linear Algebra and its Applications* 58, 109–124.

Grubisic, I. and R. Pietersz (2004) “Efficient Rank Reduction of Correlation Matrices”, Working Paper Series, *SSRN*, http://ssrn.com/abstract=518563.

Gutowski, M. (2001) “Lévy flights as an underlying mechanism for global optimization algorithms”, arXiv:math-ph/0106003v1. http://arxiv.org/abs/math-ph/0106003v1.

Helton, J.W., S. Pierce, and L. Rodman (1989) “The Ranks of Extremal Positive Semidefinite Matrices with given Sparsity Pattern” *SIAM Journal on Matrix Analysis and its Applications* 10, 407–423.

Heston, S.L. (1993) “A Closed-form Solution for Options with stochastic Volatility with Applications to Bond and Currency Options” *The Review of Financial Studies* 6, 327–343.

Higham, N.J. (2002) “Computing the Nearest Correlation Matrix – A Problem from Finance” *IMA Journal of Numerical Analysis* 22, 329-343.

Holland, J. (1975) *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press: Ann Arbor.

Johnson, C.R. (1990) “Matrix Completion Problems: A Survey” in *Matrix theory and applications*, *Proc. Sym. App. Math., American Mathematical Society* vol. 40 by C.R. Johnson, Ed., 171-198.

Kahl, C. and M. Günther (2005). “Complete the Correlation Matrix” http://www.math.uni-wuppertal.de/~kahl/publications/CompleteTheCorrelationMatrix.pdf

Karaboga, D. and B. Basturk (2007) “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm” *Journal of Global Optimization* 39 (3), 459-471.

Karaboga, D. (2005) “An idea based on honey bee swarm for numerical optimization” *Technical Report-TR06*, Computer Engineering Department, Erciyes University: Turkey.

Kirkpatrick, S., C.D. Gelatt Jr., and M.P. Vecchi (1983) "Optimization by Simulated Annealing" S*cience*, 220 (4598), 671-680.

Laloux, L., P. Cizeau, M. Potters and J.P. Bouchaud (1999) "Random Matrix Theory and Financial Correlations", Science & Finance (CFM) working paper archive 500053, *Science & Finance, Capital Fund Management*. (Mathematical Models and Methods in Applied Sciences) http://www.math.nyu.edu/faculty/avellane/LalouxPCA.pdf.

Laurent, M. (2001) "Matrix Completion Problems" *The Encyclopedia of Optimization* 3, 221–229.

Lotem, A., H. Nakamura and A. Zahavi (1995) "Constraints on egg discrimination and cuckoo–host co-evolution" *Animal Behav*, 49(5), 1185–1209.

Lŭcíc, P. and D. Teodorovíc (2001) "Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence", in *Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis*, Sao Miguel: Azores Islands, 441–445.

Markowitz, H. (1952) "Portfolio Selection" *The Journal of Finance* 7(1), 77–91.

Mishra, S.K. (2004) "Optimal Solution of the Nearest Correlation Matrix Problem by Minimization of the Maximum Norm" http://ssrn.com/abstract=573241.

Mishra, S.K. (2006a) "Global Optimization by Differential Evolution and Particle Swarm Methods: Evaluation on Some Benchmark Functions" Working Paper Series, *MPRA*. http://mpra.ub.uni-muenchen.de/1005/1/MPRA_paper_1005.pdf.

Mishra, S.K. (2006b) "Some New Test Functions for Global Optimization and Performance of Repulsive Particle Swarm Method" *Social Science Research Network (SSRN) Working Papers Series*, http://ssrn.com/abstract=927134.

Mishra, S.K. (2007a) "Minimization of Keane's Bump Function by the Repulsive Particle Swarm and the Differential Evolution Methods" MPRA Working Paper, http://mpra.ub.uni-muenchen.de/3098/1/MPRA_paper_3098.pdf

Mishra, S.K. (2007b) "Completing Correlation Matrices of Arbitrary Order by Differential Evolution Method of Global Optimization: A Fortran Program" *SSRN* http://ssrn.com/abstract=968373.

Mishra, S.K. (2010) "Performance of Differential Evolution and Particle Swarm Methods on Some Relatively Harder Multi-modal Benchmark Functions" *The IUP Journal of Computational Mathematics* III(1), 7-18.

Münnix, M.C., T. Shimada, R. Schäfer, F. Leyvraz, T.H. Seligman, T. Guhr, and H.E. Stanley (2012) "Identifying States of a Financial Market" *Scientific Reports* 2, 644, DOI: 10.1038/srep00644.

Nagy, M.E., M. Laurent, and A. Varvitsiotis (2012) "Complexity of the Positive Semidefinite Matrix Completion Problem with a Rank Constraint" http://arxiv.org/pdf/1203.6602.pdf.

Nelder, J.A. and R. Mead. (1964) "A Simplex Method for Function Minimization" *Computer Journal* 7, 308-313.

Olkin, I. (1981) "Range Restrictions for Product-Moment Correlation Matrices" *Psychometrika* 46, 469-472.

Ormerod, P. and C. Mounfield (2000) "Random Matrix Theory and the Failure of Macro-economic Forecasts" http://arxiv.org/ftp/cond-mat/papers/0102/0102357.pdf.

Pavlyukevich, I. (2007) "Lévy flights, non-local search and simulated annealing" *J. Computational Physics* 226, 1830-1844.

Pietersz, R and P.J.F. Groenen (2004) "Rank Reduction of Correlation Matrices by Majorization", *Econometric Institute Report* EI 2004-11, Erasmus Univ.: Rotterdam.

Potters, M., J.P. Bouchaud and L. Laloux (2005) "Financial Applications of Random Matrix Theory: Old Laces and New Pieces" *Acta Physica Polonica-B* 36(9), 2767–2784. Available at http://arxiv.org/pdf/physics/0507111v1.pdf.

Rajabioun, R. (2011) "Cuckoo Optimization Algorithm" *Applied Soft Computing* 11, 5508–5518.

Rebonato, R and P. Jäckel (1999) "The Most General Methodology to Create a Valid Correlation Matrix for Risk Management and Option Pricing Purposes" *Quantitative Research Centre*, NatWest Group, http://www.rebonato.com/CorrelationMatrix.pdf.

Rothstein, S.I. (1990) "A model system for co-evolution: avian brood parasitism" *A. Rev. Ecol. Syst*. 21, 481–508.

Schöbel, R. and J. Zhu (1999). "Stochastic Volatility With an Ornstein Uhlenbeck Process: An Extension" *European Finance Review* 3, 23–46, ssrn.com/abstract=100831.

Stanley, J and M. Wang (1969) "Restrictions on the Possible Values of r12, given r13 and r23" *Educational and Psychological Measurement* 29, 579-581.

Storn, R. and K. Price (1995) "Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces" *Technical Report*, International Computer Science Institute: Berkley.

Teodorović, D., T. Davidović, and M. Šelmić (2011) "Bee Colony Optimization: The Applications Survey" http://www.mi.sanu.ac.rs/~tanjad/BCO-ACM-Trans-Ver2.pdf.

Törn, A.A. and S. Viitanen (1994) "Topographical Global Optimization using Presampled Points" *J. of Global Optimization* 5, 267-276.

Tsallis, C. and D.A. Stariolo (1995) "Generalized Simulated Annealing" ArXive condmat/9501047 v1 12 Jan.

Tumminello, M., F. Lillo and R.N. Mantegna (2010) "Correlation, Hierarchies, and Networks in Financial Markets" *J. of Economic Behavior & Organization* 75, 40-58.

Tyagi, R and C. Das (1999) "Grouping Customers for Better Allocation of Resources to Serve Correlated Demands" *Computers and Operations Research* 26, 1041-1058.

Valian, E., E. Mohanna and S. Tavakoli (2011) "Improved Cuckoo Search Algorithm for Global Optimization" *Int. J. Communications and Information Technology* 1(1), 31-44.

Viswanathan, G. M., V. Afanasyev, S.V. Buldyrev, E.J. Murphy, P.A. Prince and H.E. Stanley (1996) "Lévy Flight Search Patterns of Wandering Albatrosses" *Nature* 381, 413–415.

Viswanathan, G. M., F. Bartumeus, S.V. Buldyrev, J. Catalan, U.L. Fulco, S. Havlin, M.G.E. da Luz, M.L. Lyra, E.P. Raposo and H.E. Stanley (2002) "Lévy Flight Random Searches in Biological Phenomena" *Physica - A* 314, 208–213.

Viswanathan, G. M., S.V. Buldyrev, S. Havlin, M.G.E. da Luz, E.P. Raposo and H.E. Stanley (1999) "Optimizing the Success of Random Searches" *Nature* 401, 911–914.

Xu, K. and P. Evers (2003) "Managing Single Echelon Inventories through Demand Aggregation and the Feasibility of a Correlation Matrix" *Computers and Operations Research* 30, 297-308.

Yang, X. S. and S. Deb (2009) "Cuckoo search via Lévy flights" *Proceeings of World Congress on Nature & Biologically Inspired Computing* (NaBIC 2009, India), IEEE Publications: USA, 210-214.

Yang, X.S. and S. Deb (2010) "Engineering Optimisation by Cuckoo Search" *Int. J. Mathematical Modelling and Numerical Optimisation* 1(4), 330–343.