Optimistic Long Term Short Term Algorithm For Numerical Optimization

Final year Project II under Machine Design Specialisation

By Pranesh Kumar Saha
Class: BMEIV Sec:B
Roll No.: 001511201113

Guide
Prof. Tapan Kumar Barman

Abstract: In this project we introduce a new hybrid heuristic optimization algorithm inspired from human memory model and cellular interaction called optimistic long term short term algorithm(OLSTM).Then the algorithm is tested against 50 benchmark functions and 5 engineering problems and compared with various results from literature.

Introduction

In recent years metaheuristic algorithms have been used as primary techniques for obtaining the optimal solutions of real engineering design optimization problems [6–9]. Such algorithms mostly benefit from stochastic operators [10] that make them distinct from deterministic approaches. A deterministic algorithm [11–13] reliably determines the same answer for a given problem with a similar initial starting point. However, this behaviour results in local optima entrapment, which can be considered as a disadvantage for deterministic optimization techniques [14]. Local optima stagnation refers to the entrapment of an algorithm in local solutions and consequently failure in finding the true global optimum. Since real problems have extremely large numbers of local solutions, deterministic algorithms lose their reliability in finding the global optimum. Nature-inspired meta-heuristic algorithms solve optimization problems by mimicking biological or physical phenomena. They can be grouped in three main categories (see Fig. 1): evolution- based, physics-based, and swarm-based methods. Evolution-based methods are inspired by the laws of natural evolution. The search process starts with a randomly generated population which is evolved over subsequent generations. The strength point of these methods is that the best individuals are always combined together to form the next generation of individuals. This allows the population to be optimized over the course of generations.

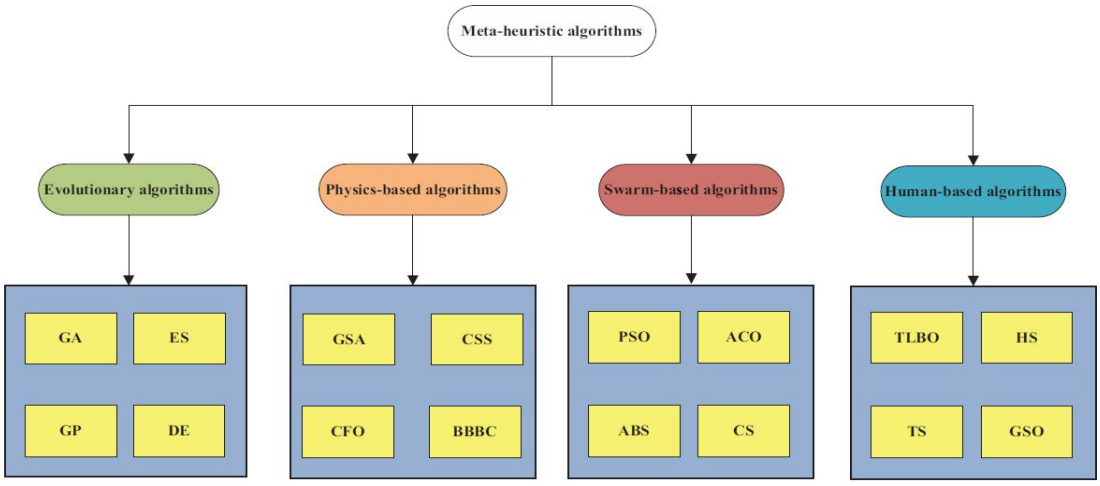


Fig. 1. Showing different classes of meta-heuristic algorithm

Evolutionary algorithms search for the global optimum in a search space by creating one or more random solutions for a given problem.This set is called the set of candidate solutions. The set of candidates is then improved iteratively until the satisfaction of a terminating condition. The improvement can be considered as finding a more accurate approximation of the global optimum than the initial random guesses. This mechanism brings evolutionary algorithms several intrinsic

advantages: problem independency,derivation independency, local optima avoidance, and simplicity. Problem and derivation independencies originate from the consideration of problems as a black box. Evolutionary algorithms only utilize the problem formulation for evaluating the set of candidate solutions. The main process of optimization is done completely independent from the problem and based on the provided inputs and received outputs. Therefore, the nature of the problem is not a concern, yet the representation is the key step when utilizing evolutionary algorithms.

Another nature-inspired methods includes swarm-based techniques that mimic the social behavior of groups of animals. The most popular algorithm is Particle Swarm Optimization, originally developed by Kennedy and Eberhart . PSO is inspired by the social behavior of bird flocking. It uses a number of particles (candidate solutions) which fly around in the search space to find the best solution (i.e. the optimal position). Meanwhile, they all trace the best location (best solution) in their paths. In other words, particles consider their own best solutions as well as the best solution the swarm has obtained so far. Another popular swarm-based algorithm is Ant Colony Optimization, first proposed by Dorigo et al. This algorithm is inspired by the social behavior of ants in an ant colony. In fact, the social intelligence of ants in finding the closest path from the nest and a source of food is the main inspiration of this algorithm. A pheromone matrix is evolved over the course of iteration by the candidate solutions.

Another important category of meta- heuristic methods are inspired by human behaviors. Some of the most popular algorithms are Teaching Learning Based Optimization(TLBO) , Harmony Search (HS) , Tabu (Taboo) Search (TS) , Group Search Optimizer (GSO) , Imperialist Competitive Algorithm (ICA) , League Championship Algorithm (LCA) , Firework Algorithm, Colliding Bodies Optimization (CBO) , Interior Search Algorithm (ISA), Mine Blast Algorithm (MBA), Soccer League Competition (SLC) algorithm, Seeker Optimization Algorithm (SOA) , Social-Based Algorithm (SBA) , Exchange Market Algorithm (EMA) , and Group Counseling Optimization (GCO) algorithm. All the human inspired algorithms try to mimic human social interaction and decision making procedure.

Population-based meta-heuristic optimization algorithms share a common feature regardless of their nature. The search process is divided into two phases: exploration and exploitation. The optimizer must include operators to globally explore the search space: in this phase, movements (i.e. perturbation of design variables) should be randomized as most as possible. The exploitation phase follows the exploration phase and can be defined as the process of investigating in detail the promising area(s) of the search space. Exploitation hence pertains to the local search capability in the promising regions of design space found in the exploration phase. Finding a proper balance between exploration and exploitation is the most challenging task in the development of any metaheuristic algorithm due to the stochastic nature of the optimization process.

In this paper we introduce a new algorithm that falls in the hybrid class of human inspired swarm based algorithm. Optimistic Long Term Short Term Memory (OLSTM) optimization algorithm

proposed is inspired from the human memory and modelling of cellular interaction representing the same.

Optimistic Long Term Short Term Memory Optimised (OLSM)

In this section we first discuss the inspiration behind the proposed algorithm and then elaborate on the mathematical model of the same.

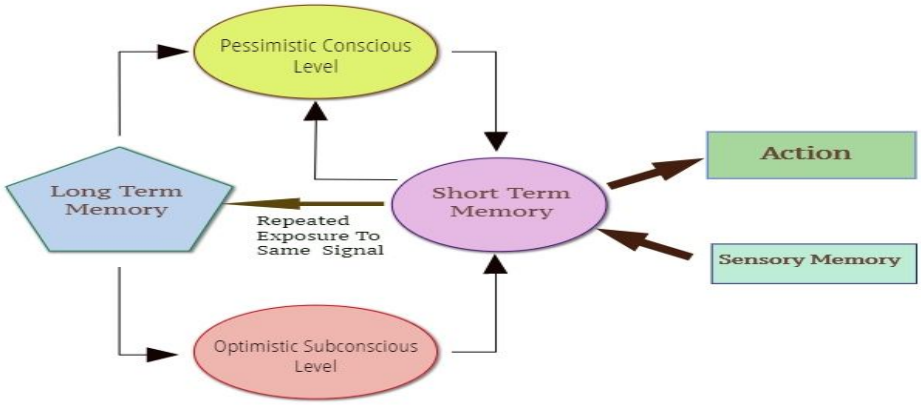


Fig 1. Memory Model

1. The Human Memory

Humans are defined specifically by their memory and in the society we are nothing but a reflection of our memory and knowledge. The fluid intelligence that we possess is also strongly correlated to our working memory(STM) [2,3] by many researches. Singularly humans are very efficient in optimising for personal gains. Although in course of evolution we have not developed any instrument to act as a swarm similar to less cranially developed life forms, many group activities are performed by human beings which are similar to swarms[5] and researches on artificial human swarms [4] is starting to pick pace.

The structure and working human brain has intrigued many great “brains” for centuries now. One of the most asked question is how exactly the human brain remembers. Over the years our understanding of ‘memory’ has developed. In 1890 the first notable study of the memory was made by William James. Based on his research he proposed a dual memory model (dichotomous memory):

- **Primary memory,** which lasts for a few second and holds information in our consciousness
- **Secondary memory,** which has unlimited duration and can be brought to consciousness if desired

Later in 1968 a more influential and well accepted memory model was proposed by Richard Atkinson and Richard Shiffrni in [1]. Their model called **Multi-store model** consisted of long-term and working or short-term memory model and was later improved by an additional

component, the sensory memory. Sensory memory contains one register for each sense and serves as an short lasting bufferzone before the information can enter short-term memory. The structure of the long-term and short-term memory proposed was widely accepted.In STM and LTM the basic differences are as follows:

- **capacity** (small for STM and large or unlimited for LTM),
- **duration limits** (items in STM decay as a function of time, which is not a characteristic of LTM),
- **retention speed** (very high for STM and possibly lower for LTM),
- **time to acquire information** (short for STM and longer for LTM),
- **information encoding (**semantic for LTM and acoustic or visual for STM), and
- **type of memory** affected by physical injuries in patients

Two important differences are the capacity and duration limits. Short term memory usually has the capability to hold about seven items for a duration of 10-30 secs. Long term memory doesn't has any such limits on its capacity and duration. The decay in memory is a very interesting character. Many decay theories have been proposed over time.The most elaborate decay-based theory of working memory to date is the "time-based resource sharing model.This theory assumes that representations in working memory decay unless they are refreshed. Refreshing them requires an attentional mechanism that is also needed for any concurrent processing task. When there are small time intervals in which the processing task does not require attention, this time can be used to refresh memory traces. The theory therefore predicts that the amount of forgetting depends on the temporal density of attentional demands of the processing task—this density is called "cognitive load". After repeated refreshing of a task or information it moves to the long term memory. Thus there is a interaction between the two memories.The OLSTM algorithm is mainly inspired from this finite storage and decay characteristic shown by the human memory.

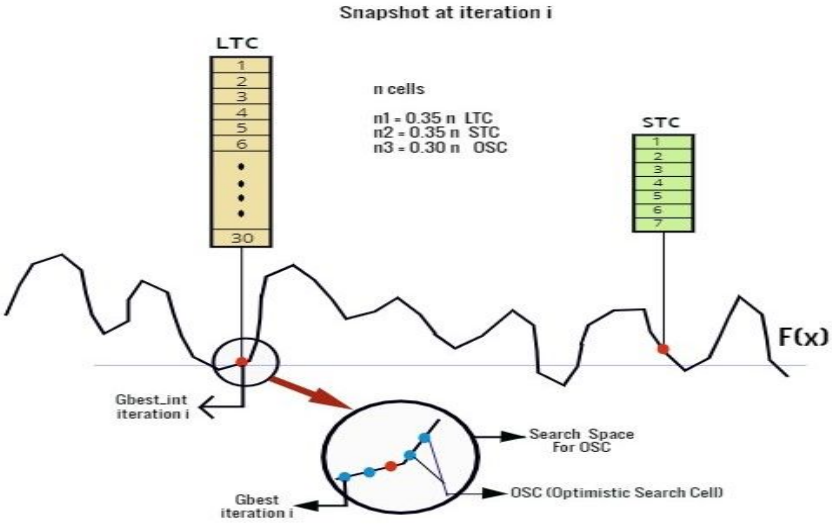


Fig 2. The Modelling Of Memory

2. Basic Model

The basic structure of the OLSTM algorithm tries to capture the model of the human memory. The basic units of the structure are called cells, which are of three types:

1. Short term memory cells (STC)

2. Long term memory cells (LTC)
3. Optimistic search cells (OSC)

The short-term cells model the short term memory process. Each of them have the following characteristics:

1. Memory: storage capacity of seven values. Modeled by an array with latest value being stored in the modulo 7 of the current iteration.

$$Position_{cur} = Iteration\%7 \quad \textbf{(1a)}$$

2. Selection function: This function selects the observable(working) memory to be used by the cell to find its personal best position.

$$Upper = random.uniform(0, Position_{cur}) \quad \textbf{(1b)}$$

where random.uniform denotes a random number generator from a uniform distribution.

$$Working_{memory} = Memory[:Upper] \quad \textbf{(2)}$$

3. Decay function: Decay occurs when the memory unit has become filled i.e after every 7th iteration. The decay function deletes all but the best position which is stored in the first position of memory.

The long-term cells model the long term memory process. They also have similar characteristics as STCs but with little different definitions:

1. Memory: storage capacity of 30 values. Modeled b by an array with latest value being stored in the modulo 30 of the current iteration.

$$Position_{cur} = Iteration\%30 \quad \textbf{(3)}$$

As we can understand over a cycle of 30 iteration all values get renewed.

2. Selection function: This function selects the observable(working) memory to be used by the cell to find its personal best position.

$$Upper = random.uniform(0, 30) \quad \textbf{(4)}$$

where random.uniform denotes a random number generator from a uniform distribution.

$$Working_{memory} = Memory[:Upper] \quad \textbf{(5)}$$

3. Reinitialization function: A reinitialization function is defined to increase the exploratory properties of the LTCs. In every 6th iteration one past position is replaced by a random new position within the space of discourse.

$$index = random.integer.uniform(0, 30) \quad \textbf{(6)}$$

$$Memory[index] = random.uniform(Low, High) \quad \textbf{(7)}$$

where Low and High represent the lower and upper bounds on each variable.

The optimistic search cells are inspired from the Pollyanna Principle.The **Pollyanna principle** (also called Pollyannaism or positivity bias) is the tendency for people to remember pleasant items more accurately than unpleasant ones.Research indicates [ ]that at the subconscious level, the mind has a tendency to focus on the optimistic.

OSCs are defined to search in a small zone around the current global best position for a better solution,i.e, they are optimistic that true optimum exists near the current optimum.

The structure and process of introduction is explained in the next section.

One important aspect is the distribution of the number of cells of each type for a given total population size. Although we were not able to establish any empirical formula for the same, various combinations were tried out on 5 different functions and the following combination was found to be the best one.

If total number of cells is n then:

$$N_{STC} = 0.35 * n \quad \textbf{(8)}$$
$$N_{LTC} = 0.35 * n \quad \textbf{(9)}$$
$$N_{OSC} = 0.30 * n \quad \textbf{(10)}$$

Now as we have discussed the important components,the process of optimization goes by the following steps:

- The STCs and LTCs are initialized in the search space randomly
- During each iteration the working memory is calculated for every current cell
- Each cell updates its velocity by cosine projection
- The minimum value obtained for each group is obtained and the intermittent global minimum is found
- The OSCs are introduced at the region around the global minimum
- The overall global minimum is found out and OSCs are terminated
- Memory update (decay and reinitialization) is carried out at the specific iterations
- Memory Ghosting

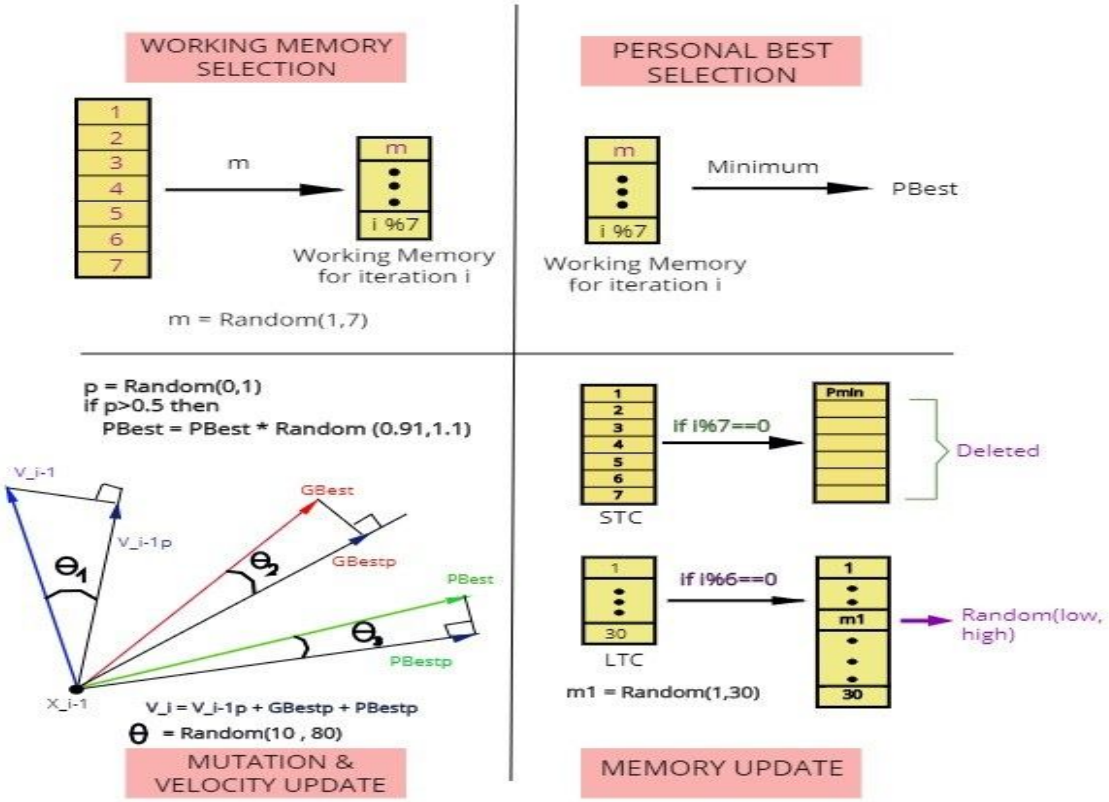


Fig 3. The Main Functions

2.1 Cosine Projection Velocity Update

In most heuristic search based optimisation algorithms the movement of the particles (cells) are either by random walk or by velocity update. In OLSTM a new method of velocity update is introduced by changing the plane of velocities.The velocity update has three contributing parts:

1. Social component: tendency of each cell to move towards the global best solution so far

$$V_{global} = (Position_{global\ best} - Position_{current}) \quad \textbf{(11)}$$

2. Personal component: tendency of each particle to move towards the personal best solution from the working memory

$$V_{personal} = (Position_{working\ memory\ best} - Position_{current}) \quad \textbf{(12)}$$

3. Previous iteration component: tendency to move with the previous velocity $V_{previous}$

Consider the figure above, in which the point O denotes the current position of the particle , and $V_{global}, V_{personal}, V_{previous}$ denotes the vectors of the velocities.For any vector the cosine of an angle times the vector denotes the orthogonal projection of the vector which can vary from 0 (perpendicular or angle=90) to 1 (parallel or angle=0) times the original vector in magnitude. This also denotes a change in the direction of motion, from the same direction to and orthogonal direction. We utilise these two properties to increase the exploratory characteristics of velocity update.

$$V_{particle} = w_1 * cos\ \theta_1 * V_{global} + w_2 * cos\ \theta_2 * V_{personal} + w_3 * cos\ \theta_3 * V_{previous} \quad \textbf{(13)}$$

where θ_1 , θ_2 and θ_3 denote the angle of projection of the global,personal and previous velocity components respectively,and

w_1 , w_2 and w_3 denotes the scaling weights defined in literature as 1.496180, 1.496180 and .729844,

The values of θ_1 , θ_2 and θ_3 are obtained in degrees by drawing random values between 10 to 80 from a uniform distribution. The range 0 to 84 is selected as cos 84 degrees is almost equal to 0.1. Going lower than 0.1 poses the risk of vanishing contribution from one or more components and thus very small velocity magnitude.

2.2 Introduction and termination of OSCs

OSCs are zero memory cells in our algorithmic definition. They don't possess any history of previous position, value or velocity. OSCs are like flickers of light that come to existence, check the value of their position and vanish.

One major phase in the OLSTM algorithm is the introduction of OSCs near the global optimum during each iteration. Two important points to decide for the OSCs are:

1. The zone of existence
2. The duration of existence

The zone of existence will define the space around the intermittent global optimum in which the OSCs will be constrained to move. As they don't have any memory their position is defined completely from the intermittent global optimum.

We defined a gradually shrinking zone by utilizing the iteration number as a parameter.

$$d = 1/(iteration)^{1/3} \quad \textbf{(14)}$$

$$Position_i = Gbest_i + random.uniform(-d * Gbest_i\ ,\ d * Gbest_i) \quad \textbf{(15)}$$

where d is the reduction coefficient and iteration is the current number of iteration

$Gbest_i$ denotes the position of the i^{th} variable of the Gbest position

random.uniform denotes a equi-probability random number generator

The above definition of zone ensures convergence and also helps in scaling the movement with the value of $Gbest_i$.

The duration of existence decides the number of iterations for which the OSCs will search around the same intermittent global optimum. As the particles have no memory ideally it should be 1. But to aid a more optimistic exploitation we kept the duration to be 2.

2.3 Memory Ghosting

Memory ghosting is carried out in the STCs after the personal best is found from the working memory. This process resembles the mutation step in evolution based algorithms and it occurs only if a random number drawn from a range meets a certain specified condition.

$$prob = random.uniform(0, 1) \quad \textbf{(16)}$$

$$if\ prob > 0.5 \quad pBest = pBest * random.uniform(.91, 1.1) \quad \textbf{(17)}$$

where prob denotes a random number between 0 and 1. So the ghosting operation has 50 percent chance of occuring in each iteration for each cell.
In the operation a number between .91 and 1.1 is multiplied with the personal best vector in the working memory. This step aids in similar way to mutation by providing a more exploratory opportunity.

OLSTM Algorithm

With the main operations defined in the previous sections now we can define the OLSTM algorithm.
OLSTM algorithm consists of 4 different classes : STC,LTC,OSC and Optimizer.
STC and LTC classes have completely similar construct. Both are defined having 5 functions: Initialize, Pbest_get, Memory_update,Position_update and Velocity_update.
Let $X_{i\,(n\,x\,m)}$ be a matrix containing n cells with m variables of optimization
$V_{i\,(n\,x\,m)}$ be a matrix of velocity of each cell
$F_{n\,x\,1}$ be a matrix of fitness values corresponding to n cells
$ST_{7\,x\,m}$ be the memory matrix of short term cells
$LT_{30\,x\,m}$ be the memory matrix of long term cells

$Gbest_{i-1}$ be global best position from previous iteration
$Gbest_i$ be global best position of this iteration

(18) $\varphi \xrightarrow{Initialize} X_i, V_i$

(19) $X_i \xrightarrow{Function} F_i$

(20) $ST[i\,\%\,7] = X_i$
$LT[i\,\%\,30] = X_i$

(21) $i, ST \xrightarrow{Pbest\ Get} X_{pbest}$

(22) $i, LT \xrightarrow{Pbest\_Get} X_{pbest}$

(23) $X_{gbest}, X_{pbest} \xrightarrow{Velocity\_Update} V_{i+1}$

(24) $V_{i+1} \xrightarrow{Position\_update} P_{i+1}$

The function *Initialize* randomly initialises position and velocity for each cell before the start of iteration (18), *Function* denotes the function to be optimized so that fitness values are obtained on applying it to X_i . *Pbest_get* obtains the personal best from the working memory based on the

iteration number and current memory array. *Velocity_update* updates the velocity of the cells by taking global best of previous iteration, personal best from *Pbest_get* and previous position from memory. *Position_update* updates the position based on the updated velocity.
Below is the pseudocode for the proposed OLSTM algorithm.

OLSTM OPTIMIZATION ALGORITHM

Initialize n1 STC, n2 LTC randomly
Calculate fitness of the STC and LTCs
Find the interim best cell for iteration 0 $Gbest_{i=0}$

while *end criterion not satisfied*
for *n1 short-term cells and n2 long-term cells*
Update memory of STC
Obtain pBest from working memory
Update velocity
Update position
Calculate fitness value for new position
end for
Find interim global best from fitness value
Calculate space of discourse for OSCs
Introduce n3 OSCs
for *n3 OSCs calculate the fitness value*
end for
Find the final global best for current iteration
end while
return *global best position found*

Hypothesis about the OLSTM algorithm

- Exploration of the search space is guaranteed by random initialization of cells, working memory selection, threshold based scaling of personal best and finally by reinitialization of random memory space of the LTCs. The combined effect of all these result in a much better exploration than previous algorithms.
- Exploitation of the search space is guaranteed by the optimistic search cells (OSC) and their shrinking space of travel.Also the two iteration duration of existence of the OSCs ensure proper exploitation of each global best position.
- There is high probability of avoiding and resolving local optimum stagnation due to memory ghosting and reinitialization.
- OLSTM is population based human inspired algorithm,which also makes avoidance of local optima intrinsically high.
- The OLSTM algorithm ensure convergence by making cells follow the global best and reducing the zone of travel of OSCs.
- Every dimension of the cells are randomised ensuring proper diversity among the cells.

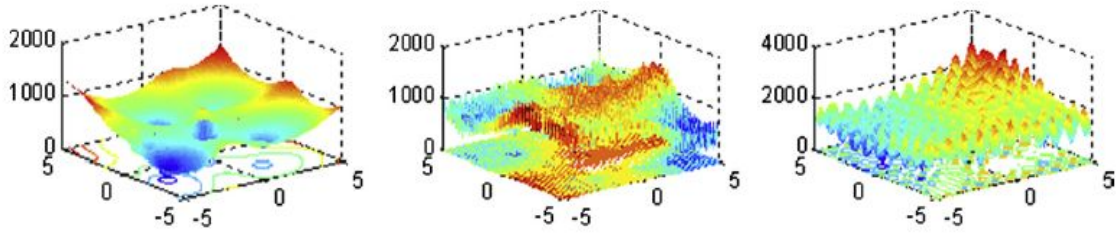
3.Results and Discussion

The OLSTM algorithm is tested on 53 functions. Of these 36 are unconstrained benchmark problems, 13 are constrained benchmark problems and 3 are engineering design optimization.

Real systems (problems) are mostly constrained. There are two types of constraints involved in defining the feasibly of solutions during the design process: inequality and equality constraints. For optimizing constrained problems, a constraint handling
method should be integrated to the optimizer. There are several methods of constraints handling in the literature: penalty functions, special operators, repair algorithms, separation of objectives and constraints, and hybrid methods. Since finding a good
constraints handling method for the proposed OLSTM algorithm is out of the scope of this work, the simplest method called death penalty is employed. In this method, search agents that violate any of the constraints with any level are treated as the same and
penalized by assigning a large fitness value (small objective value in case of maximization). This method is very cheap and readily applicable for the OLSTM algorithm without algorithm modifications.

In unconstrained problems there are 3 classes of benchmark problems: unimodal, multimodal and composite. Unimodal test functions have single optimum so they can benchmark the exploitation and convergence of an algorithm. In contrast, multi-modal test functions have more than one optimum, making them more challenging
than unimodal functions. One of the optima is called global optimum and the rest are called local optima. An algorithm should avoid all the local optima to approach and determine the global optimum. Therefore, exploration and local optima avoidance of algorithms can be tested by the multi-modal test functions. Note that the minima of all the unimodal and multimodal test functions are equal to 0 except the function F8 where the minimum is changed based of the number of variables (Dim).

The last group of test functions, composite functions, are mostly the combined, rotated, shifted, and biased version of other unimodal and multi-modal test functions. As shown in figure below, they mimic the difficulties of real search spaces by providing a massive number of local optima and different shapes for different regions of the search space. An algorithm should properly balance exploration and exploitation to approximate the global optima of such test functions. Therefore, exploration and exploitation combined
can be benchmarked by this group of test functions.



C1 C2 C3

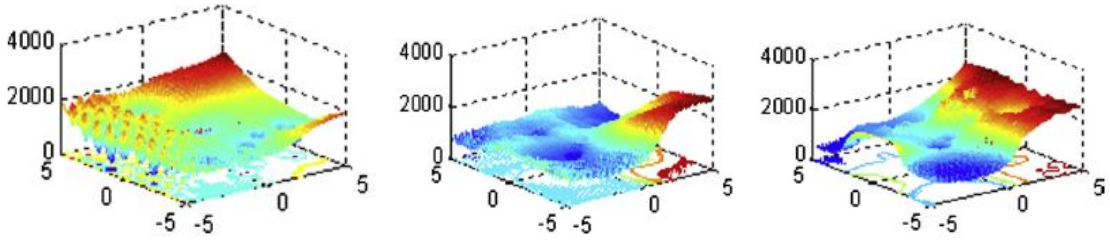


Fig.4. Graphs showing C1,C2,C3,C4,C5,C6

For all problems the number of cells is taken as 200 and number of iteration is limited to 2000. Average value of minimum in 30 runs and their std is reported.

Name	Unimodal Function	Dim	Range	Min
F1	$\Sigma^{n}_{i=1}x^{2}_{i}$	30	[-100,100]	0
F2	$\Sigma^{n}_{i=1}\|x_i\| + \prod^{n}_{i=1}\|x_i\|$	30	[-10,10]	0
F3	$\Sigma^{n}_{i=1}(\Sigma^{n}_{j=1}x_j)^2$	30	[-100,100]	0
F4	$max_i\{\|x_i\|, 1 <= i <= n\}$	30	[-100,100]	0
F5	$\Sigma^{n}_{i=1}[100(x_{i+1} - x^{2}_{i})^2 + (x_i - 1)^2]$	30	[-30,30]	0
F6	$\Sigma^{n}_{i=1}([x_i + 0.5])^2$	30	[-100,100]	0
F7	$\Sigma^{n}_{i=1}ix^4 + random[0,1)$	30	[-1.28,1.28]	0
F8	$\Sigma^{n}_{i=1}ix^{2}_{i}$	30	[-10,10]	0
F9	$\Sigma^{n}_{i=1}(1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	5	[-4.5,4.5]	0
F10	$-cos(x_1)cos(x_2)exp(-(x_1 - 3.14)^2 - (x_2 - 3.14)^2))^2$	2	[-100,100]	-1
F11	$0.26(x_1^2 + x_2^2) - .48x_1x_2$	2	[-10,10]	0
F12	$100(x_1^2 - x_2^2) + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4) + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) - 0.48x_1x_2 + 19.8(x_2 - 1)(x_4 - 1)$	4	[-10,10]	0
F13	$\Sigma^{n}_{i=1}(x_i - 1)^2 - \Sigma^{n}_{i=2}x_ix_{i-1}$	10	$[-n^2, n^2]$	-210

F14	$\Sigma^{n}_{i=1}x_i^2 + (\Sigma^{n}_{i=1}0.5ix_i)^2 + (\Sigma^{n}_{i=1}0.5ix_i)^4$	10	[-5,10]	0
F15	$(x_1 - 1)^2 + \Sigma^{n}_{i=2}i(2x_i^2 - x_{i-1})^2$	30	[-30,30]	0
F16	$[1/500 + \Sigma^{25}_{j=1}(1/(j + \Sigma^{2}_{i=1}(x_i - a_{ij})^6)]^{-1}$	2	[-65.536,65.536]	.998

Table 1. Unimodal Functions

sl.n	Multimodal Function	dim	Range	Min
M1	$\Sigma^{n}_{i=1} - x_i sin(\sqrt{\lvert x_i \rvert})$	30	[-500,500]	-418.9829 x30
M2	$\Sigma^{n}_{i=1}[x_i^2 - 10cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
M3	$-20exp(-0.2\sqrt{(1/n\Sigma^{n}_{i=1}x_i^2)} - exp(1/n\Sigma^{n}_{i=1}cos(2\pi x_i)) + 20 + e$	30	[-32,32]]	0
M4	$1/4000\Sigma^{n}_{i=1}x_i^2 - \prod^{n}_{i=1}cos(x_i/\sqrt{i}) + 1$	30	[-600,600]	0
M5	$\pi/n\{10sin(\pi y_1) + \Sigma^{n-1}_{i=1}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \Sigma^{n}_{i=1}u(x_i, 10, 100, 4)$ $y_i = 1 + (x_i + 1)/4$ $u(x_i, a, k, m) = k(x_i - a)^m \quad x_i > a$ $0 \quad -a < x_i < a$ $k(-x_i - a)^m \quad x_i > a$	30	[-50,50]	0
M6	$0.1\{sin^2(3\pi x_1) + \Sigma^{n}_{i=1}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)]\} + + \Sigma^{n}_{i=1}u(x_i, 5, 100, 4)$	30	[-50,50]	0
M7	$(x_2 - (5.1/4\pi^2)x_1^2 + (5/\pi)x_1 - 6)^2 + 10(1 - 1/8\pi)cos\, x_1 + 10$	2	[-5,10] [0,15]	0
M8	$x_1^2 + 2x_2^2 - 0.3cos(3\pi x_1) - 0.4cos(4\pi x_2) + 0.7$	2	[-100,100]	0
M9	$x_1^2 + 2x_2^2 - 0.3cos(3\pi x_1)cos(4\pi x_2) + 0.3$	2	[-100,100]	0
M10	$x_1^2 + 2x_2^2 - 0.3cos(3\pi x_1 + 4\pi x_2) + 0.3$	2	[-100,100]	0
M11	$-\Sigma^{n}_{i=1}sinx_i(sin(ix_i^2/\pi))^{20}$	5	[0,π]	-4.687
M12	$[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] * [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2]$	2	[-2,2]	3
M13	$\Sigma^{n}_{k=1}[\Sigma^{n}_{i=1}(i^k + 0.5)((x_i/i)^k - 1)]^2$	4	[-4,4]	0

Table 2. Multimodal Functions

sl.no	Composite function	dim	range	min
C1	$f_1,f_2,.....,f_{10}$ = Sphere Function $[\sigma_1,\sigma_2,....\sigma_{10}]=[1,1,.....,1]$ $[\lambda_1,\lambda_2,.....,\lambda_{10}]= [5/100,5/100,.....,5/100]$	30	[-5,5]	0
C2	$f_1,f_2,.....,f_{10}$ = Griewank's Function $[\sigma_1,\sigma_2,....\sigma_{10}]=[1,1,.....,1]$ $[\lambda_1,\lambda_2,.....,\lambda_{10}]= [5/100,5/100,.....,5/100]$	30	[-5,5]	0
C3	$f_1,f_2,.....,f_{10}$ = Griewank's Function $[\sigma_1,\sigma_2,....\sigma_{10}]=[1,1,.....,1]$ $[\lambda_1,\lambda_2,.....,\lambda_{10}]=[1,1,.....,1]$	30	[-5,5]	0
C4	f_1,f_2= Ackley's fnc, f_3,f_4= Rastrigin's fnc. , f_5,f_6= Weierstrass fnc., f_7,f_8 = Griewank's fnc, f_9,f_{10} = Sphere fnc $[\sigma_1,\sigma_2,....\sigma_{10}]=[1,1,.....,1]$ $[\lambda_1,\lambda_2,.....,\lambda_{10}]= [5/32,5/32,1,1,5/.5,5/.5,5/100,5/100,5/100,5/100]$	30	[-5,5]	0
C5	f_1,f_2= Rastrigin's fnc, f_3,f_4= Weierstrass fnc. , f_5,f_6=Griewank's fnc., f_7,f_8 = Ackley's fnc , f_9,f_{10} = Sphere fnc $[\sigma_1,\sigma_2,....\sigma_{10}]=[1,1,.....,1]$ $[\lambda_1,\lambda_2,.....,\lambda_{10}]=[1/5,1/5,5/0.5,5/0.5 ,5/100,5/100,5/32,5/32,5/100, 5/100]$	30	[-5,5]	0
C6	f_1,f_2= Rastrigin's fnc, f_3,f_4= Weierstrass fnc. , f_5,f_6=Griewank's fnc., f_7,f_8 = Ackley's fnc , f_9,f_{10} = Sphere fnc $[\sigma_1,\sigma_2,....\sigma_{10}]=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1]$ $[\lambda_1,\lambda_2,.....,\lambda_{10}]=[0.1*1/5,0.2*1/5,0.3*5/0.5,0.4*5/0.5,,0.5*5/100,0.6*5/100, 0.7*5/32,0.8*5/32,0.9*5/100,1*5/100]$	30	[-5,5]	0

Table 3. Composite Functions

In below table the results for unimodal function are given. The mean error from optimum is reported.

	OLSTM		PSO		GA	
	Mean	Std	Mean	Std	Mean	Std
F1	0	0	2.70E-09	1.00E-09	0.118842	0.01256
F2	0	0	7.15E-05	2.26E-05	0.14522	0.053227
F3	0	0	4.71E-06	1.49E-06	0.13902	0.1211
F4	0	0	3.20E-07	1.02E-08	0.15795	0.862029
F5	0.163	0.035	0.123	0.2162	0.714157	0.972711
F6	0.00358	0.0000045	5.27E-07	2.70E-06	0.167918	0.86868
F7	0.00025	1.20E-04	0.001398	0.0012	0.01007	0.003263

F8	2.51E-79	1.33E-77	1.78E-69	2.50E-68	1.48E+02	12.409
F9	6.23E-26	1.57E-25	2.12E-11	3.17E-12	3.50E-29	1.20E-28
F10	1.69E-06	1.30E-12	1.75E-05	1.89E-08	1.89E-04	2.78E-07
F11	0	0	0	0	0	0
F12	2.16E-06	2.40E-10	4.90E-04	4.78E-07	0.01493	0.0073
F13	0.0025	0.0001	3.40E-05	1.20E-2	0.1934	0.03531
F14	9.50E-150	1.50E-151	0	0.00E+00	0	0
F15	0.665	0.00012	0.6667	1.00E-08	1.22E+03	2.66E+02
F16	3.80E-06	3.75E-10	1.34E-04	0	1.47E-04	0

Table 4. Results of Unimodal Functions

The below table gives the values obtained for multimodal functions.

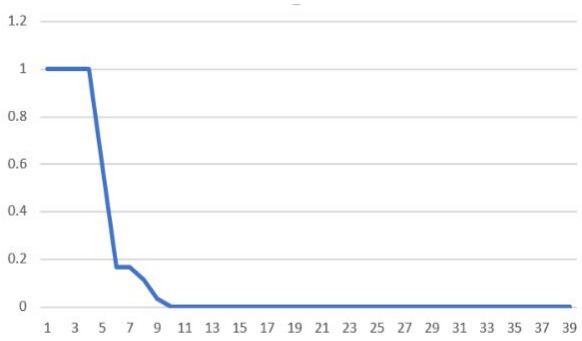
	OLSTM		PSO		GA	
	Mean	Std	Mean	Std	Mean	Std
M1	-7843.72	200.49	-1.37E+03	1.46E+02	-2091.64	2472.35
M2	0	0	2.79E-01	2.10E-01	0.65927	0.81575
M3	4.00E-15	1.20E-17	1.10E-09	2.30E-11	0.95611	0.807701
M4	0	0	2.74E-01	2.04E-01	0.487809	0.217782
M5	0.000103	0.000024	9.42E-09	2.31E-08	0.110769	0.002152
M6	0.0038	0.001	2.00E-11	2.88E-08	1.29E-01	0.06885
M7	3.94E-01	9.10E-09	3.98E-01	0	0.397887	0
M8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0
M9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
M10	-4.65E+00	0.00E+00	-2.49E+00	2.57E-01	-4.64E+00	9.79E-02
M11	0	0	0	0	0.06829	0.078216
M12	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0	0
M13	3.00E+00	2.65E-17	3.00E+00	0.00E+00	5.870093	1.07171
M14	1.20E-19	2.87E-21	0.036	4.89E-02	0.302671	0.19325

Table 5 . Results of Multimodal Functions

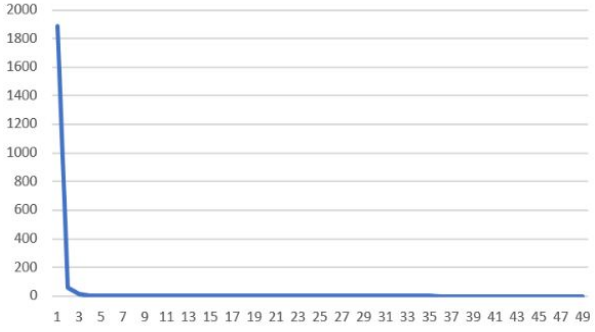
The table given below shows the results of composite benchmark functions.

	OLSTM		PSO		GA	
	Mean	Std	Mean	Std	Mean	Std
G1	3.45	1.0034	1.20E+02	1.32E+02	114.613	26.962
G2	7.89	2.0013	1.63E+02	1.19E+02	95.4633	7.1633
G3	5.76E+01	5.78E+00	3.63E+02	1.51E+02	325.4427	51.668
G4	71.134	17.498	4.50E+02	1.76E+02	466.3074	29.568
G5	12.375	0.00031	1.75E+02	1.76E+02	90.369	13.729
G6	900.789	10.419	9.02E+02	8.39E-01	5.21E+02	27.98507

Table 6. Results of Composite Functions

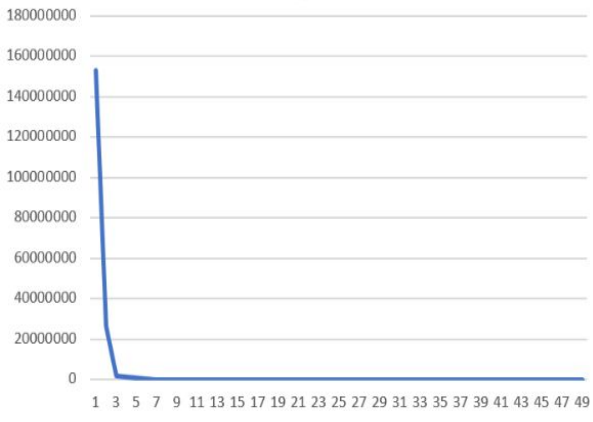


F10

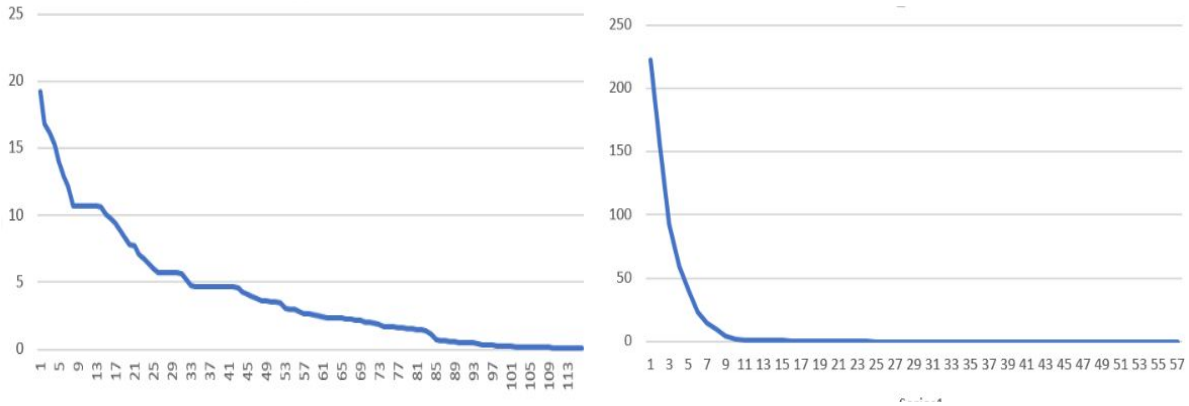


F12

F13



M5



M3 M4

Fig. 5. Graphs showing change in value of error for F10(Easom), F12(colvile),F13(Trid 10), M5(penalised fnc 1), M3 (ackeley), M4(Griewank) over iterations

From the results and graphs above the following conclusions can be drawn:

- OLSTM has a very high speed of convergence for both unimodal (F10,F12,F13) and multimodal(F21). Even when search space is large the rate of convergence is very fast.
- OLSTM outperform PSO and GA in 13 unimodal functions, 11 multimodal functions and 5 composite functions. This shows it is a much superior algorithm.
- For composite functions OLSTM shows its capability of maintaining exploration and exploitation balance by giving much better results than either GA or PSO. This implies although quick convergence is a feature of OLSTM in multimodal and unimodal functions, exploration becomes dominant in composite functions.

Constrained Optimization

Although this paper doesn't gives a new method for constrained optimization we test OLSTM's performance for constrained optimization to get a better understanding of exploratory characteristics.As mentioned earlier we use dead weight penalty technique for constrained optimization. 200 cells and 5000 iterations were used for each function.

We test against 13 benchmark functions taken from Chaotic grey wolf optimization algorithm for constrained optimization problems Mehak Kohlia, Sankalap Aroraa, paper.

	OLSTM	GWO	FA	OPTIMUM
CS1	-14.56	-14.3159	-67.6314	-15
CS2	-0.708	-0.31375	-0.51773	-0.80362
CS3	-0.9905	0.8391	-1.99369	-1
CS4	-30514.87	-33141.1	-30446.7	-30665.5
CS5	5291.78	43924.93	97119.4	5126.498
CS6	-6176.37	-6265.65	-6849.86	-6961.81
CS7	25.04	42.1324	27.654	24.30621
CS8	-0.0156	-672.078	-11.0266	-0.09583
CS9	680.556	603.816	680.438	680.6301
CS10	7049.21	6653.97	6091.5	7049.331

CS11	0.7214	0.693	0.625	0.75
CS12	-2.67	-47.36	-53.2563	-1
CS13	0.1783	1.09478	0.856731	0.0539

Table 7. Results of constrained functions

It can be seen that OLSTM gives good results for each of the problem and even outperforms specialised constrained optimization algorithms (Grey wolf and fire fly) in 7 of the functions.

Mechanical design problems

Lastly we test our algorithm against three mechanical engineering design problems to see the real world problem solving capabilities of our algorithm.

1.Tension/Compression spring

The main goal of this engineering design problem is to minimize the weight of the spring involving three decision variables which are wire diameter (d), mean coil diameter (D) and number of active coils (N). This problem is subjected to three inequality constraints and an objective function.

Consider $x = |x_1, x_2, x_3|$

Minimize $f(x) = (x_3+2)x_2x_1^2$

Subject to $g_1(x) = 1 - x_2^3x_3/(71785x_1^4) <= 0$

$g_2(x) = ((4x_2^2-x_1x_2)/(12566(x_2x_1^3-x_1^4)))+(1/5108x_1^2)<=0$

$g_3(x) = 1 - 140.45/(x_2^2x_3) <= 0$

$g_4(x) = (x_1+x_2)/1.5 -1 <= 0$

Variable range: $0.05 <= x_1 <= 2$

$0.25 <= x_2 <= 1.3$

$2.00 <= x_3 <= 15$

Algorithm	d	D	N	Optimum wt.
OLSTM	0.051207	0.345215	12.004032	0.0126763
PSO	0.051728	0.357644	11.244543	0.0126747
GA	0.051480	0.351661	11.632201	0.0126810

Table 8. Result of Spring Optimisation

2.Welded Beam Design

The objective of this test problem is to minimize the fabrication cost of the welded beam. Optimization constraints are on shear stress (τ) and bending stress in the beam (θ), buckling load (P c), end deflection of the beam (δ). There are four optimization variables: thickness of weld (h), length of the clamped bar (l), height of the bar (t), and thickness of the bar (b).

Consider $x = [x_1 x_2 x_3 x_4] = [h l t b]$,
Minimize $f(x) = 1 . 10471 x 2 1 x 2 + 0 . 04811 x 3 x 4 (14 . 0 + x 2)$,
Subject to $g_1(x) = \tau(x) - \tau_{max} \leq 0$,
$g_2(x) = \sigma(x) - \sigma_{max} \leq 0$,
$g_3(x) = \delta(x) - \delta_{max} \leq 0$,
$g_4(x) = x_1 - x_4 \leq 0$,
$g_5(x) = P - P_c(x) \leq 0$,
$g_6(x) = 0 . 125 - x_1 \leq 0$
$g_7(x) = 1.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \leq 0$
Variable range $0 .1 \leq x_1 \leq 2$, $0.1 \leq x_2 \leq 10$, $0 .1 \leq x_3 \leq 10$,$0 . 1 \leq x_4 \leq 2$
where $\tau(x) = \sqrt{ \tau^2 + 2\tau' \tau'' x_2 /(2R) + \tau^2 }$,
$\tau' = P /(\sqrt{2} x_1 x_2)$, $\tau'' = MR/J$, $M = P(L + x_2/2)$
, $R = \sqrt{ x_2^2/ 4 + ((x_1 + x_3)/2)^2 }$,
$J = 2 \{\sqrt{2} x_1 x_2 [x_2^2/ 4 + ((x_1 + x_3)/2)^2]\}$,
$\sigma(x) = 6 P L /(x_4 x_3^2)$,
$\delta(x) = 6 P L^3/(E x_3^2 x_4)$
$P_c(x) = ((4 . 013 E \sqrt{ x_3^2 x_4^6 /36)/ L^2})(1 - x_3 /(2 L)\sqrt{ (E /4 G) })$
, $P = 60 0 0$ lb , $L = 14$ in ., $\delta max = 0 . 25$ in ., $E = 30 \times 1 6$ psi , $G = 12 \times 10^6$ psi , $\tau max = 13600$ psi , $\sigma max = 30000$ psi ,

ALGORITHM	h	l	t	b	OPTIMUM COST
OLSTM	0.205396	3.484293	9.037426	0.206276	1.730499
GSA	0.182129	3.856979	10	0.202376	1.879952
CBO	0.205722	3.47041	9.037276	0.205735	1.724663

Table 9. Result of Welded Beam Optimisation

3. Pressure vessel design

In this problem the goal is to minimize the total cost (material, forming and welding) of the cylindrical pressure vessel . Both ends of the vessel are capped while the head has a hemispherical shape. There are four optimization variables: the thickness of the shell (T_s), the thickness of the head (T_h), the inner radius (R), the length of the cylindrical section without considering the head (L).
Consider $x = [x_1 x_2 x_3 x_4] = [T_s T_h R L]$,
Minimize $f(x) = 0 . 6224 x_1 x_3 x_4 + 1 . 7781 x_2 x_3^2 + 3 .1661 x_1^2 x_4 + 19 . 84 x_1^2 x_3$,
Subject to $g_1(x) = -x_1 + 0 . 0193 x_3 \leq 0$,
$g_2(x) = -x_3 + 0 . 00954 x_3 \leq 0$,
$g_3(x) = -\pi x_3^2 x_4 - 4/3 \pi x_3^3 + 1296000 \leq 0$,
$g_4(x) = x_4 - 240 \leq 0$,
Variable range $0 \leq x_1 \leq 99$, $0 \leq x_2 \leq 99$, $10 \leq x_3 \leq 200$, $10 \leq x_4 \leq 200$,

Algorithm	T_s	T_h	R	L	Optim. cost
OLSTM	0.81250	0.437500	42.098269	176.63899	6059.7410
Imprv. HS	1.125	0.62500	58.29015	43.69268	7197.730
PSO	0.812500	0.437500	42.091266	176.7465	6061.0777

Table 10. Result of Pressure Vessel Optimisation

The results show that the OLSTM algorithm is very well suited for engineering problem optimization also.

4. 3 Bar Truss

The second problem is to design a three-bar truss to minimize its weight . The objective function is very simple, yet the problem is highly constrained. The structural design problems usually have a large number of constraints. The constraints here are
stress, deflection, and buckling constraints.

Consider $x = [x_1\ x_2\ ]= [A_1\ A_2]$;

Minimize $f(x)= (2\sqrt{2}x_1+x_2)*l$

Subject to $g_1(x)=((\sqrt{2}x_1+x_2)/(\sqrt{2}x_1^2+2x_1x_2))*P-\sigma <=0$

$g_2(x)= (x_2/(\sqrt{2}x_1^2+2x_1x_2))*P-\sigma <=0$

$g_3(x)= (1/(\sqrt{2}x_2+x_1))*P-\sigma <=0$

Variable range $0<= x_1,x_2<=1$

Where l=100 cm, P = 2KN/cm^2 , σ= 2KN/cm^2

Algorithms	x_1	x_2	minimum	Max_eval
OLSTM	0.7886784987956555	0.4082387836875023	263.89584423852403	15000
PSO	0.7886751	0.7886751	263.8958433	17600
DEDS	0.78867513	0.40824828	263.8958434	15000

Table 11. Result of 3 Bar Truss Optimisation

5. Cantilever Beam

A cantilever beam includes five hollow elements with square shaped cross-section.Each element is defined by one variable while the thickness is constant, so there is a total of 5 structural parameters. There is also a vertical load applied to the free end of the beam (node 6) and the right side of the beam (node 1) is rigidly supported. The objective is to minimize the weight of the beam. There is also one vertical displacement constraint that should not be violated by the final optimal design.

Consider $x=[x_1,x_2,x_3,x_4,x_5]$,

Minimize $f(x) = 0.6224(x_1+x_2+x_3+x_4+x_5)$,

Subject to $g(x)= (61/x_1^3)+(37/x_2^3)+(19/x_3^3)+(7/x_4^3)+(1/x_5^3) <=1$,

Variable range $0.01<= x_1,x_2,x_3,x_4,x_5 <=100$

Algorithm	x_1	x_2	x_3	x_4	x_5	Opt. Wt.
OLSTM	6.00610893	5.30564078	4.50024259	3.50715027	2.15465056	13.3652888414533
SOS	6.01878	5.30344	4.49587	3.49896	2.15564	13.36460
CS	6.0089	5.3049	4.5023	3.5077	2.1504	13.36554

Table 12. Result of Cantilever Beam optimisation

Experimental Problem Solving

We solve a problem on experimental data optimization using OLSTM to further check the accuracy of our algorithm.

The experimental dataset contains 125 roughness values for milling with combinations of five different values of depth of cut, speed and feed rate. The values are coded in five discrete values and a polynomial curve is fit into the data.

Levels	Depth of Cut (d)	Speed(N)	Feed Rate (f)
-1	0.15	2500	300
-0.5	0.175	2750	350
0	0.2	3000	400
0.5	0.225	3250	450
1	0.25	3500	500

Table 13. Coding Of Values For Experimental Data

$$R(d,N,f) = 0.00490286d^2 + 0.06602286N^2 + 0.01293714f^2 - 0.029016dN - 0.03872Nf - 0.002912df + 0.142712d + 0.004808N + 0.02f + 0.59374857$$

The above equation is optimized using OLSTM.

Label	d	N	f	Roughness
ACTUAL	-1	-0.5	-1	0.426
OPTIMIZER	-0.9999970721870496	-0.5003292534427725	-0.9988535728340631	0.42589

Table 14. Actual and Optimizer Data

The roughness calculated through the fitted function is expectedly erroneous and differs from experimental values. But the point of minima remains the same.

Conclusion

In this project we have introduced a new optimization algorithm based on human memory model. The tests show that OLSTM has the following promising features:

- It has high convergence speed for wide range of problems.
- It has high exploratory capability evident from results of composite functions.
- It has good local optima avoidance tendency and escaping capability as shown by the multimodal functions.

Scope of future work lies in exploration of better methods for constrained optimization with OLSTM, multiobjective optimization with OLSTM and high dimensional problem solving capability exploration.

Literature Review

1. Atkinson, R.C. & Shiffrin, R.M. Chapter: Human memory: A proposed system and its control processes. In Spence, K.W.; Spence, J.T. The psychology of learning and motivation (Volume 2). New York: Academic Press. pp. 89–195. 1968.
2. Kane, M. J., & Engle, R. W. (2002). The role of prefrontal cortex in working-memory capacity, executive attention, and general fluid intelligence: An individual differences perspective. Psychonomic Bulletin & Review, 9(4), 637-671.
3. Kane, M. J., Hambrick, D. Z., & Conway, A. R. A. (2005). Working memory capacity and fluid intelligence are strongly related constructs: Comment on Ackerman, Beier, and Boyle (2005). Psychological Bulletin, 131(1), 66-71
4. Rosenberg, L.B., "Human Swarms, a real-time method for Collective Intelligence." Proceedings of the European Conference on Artificial Life 2015, pp. 658-659.
5. Bashyal, S., & Venayagamoorthy, G. (2008). Human swarm interaction for radiation source search and localization. In IEEE swarm intelligence symposium, 2008 (pp. 1–8)
6. Holland JH . Genetic algorithms. Sci Am 1992;267:66–72 .
7. J.R. Koza, "Genetic programming,"1992.
8. Simon D . Biogeography-based optimization. IEEE Trans Evol Comput 2008;12:702–13 .
9. Kirkpatrick S , Gelatt CD , Vecchi MP . Optimization by simmulated annealing. Science 1983;220:671–80 .
10. ˇCernýV . Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. J Opt Theory Appl 1985;45:41–51 .
11. Webster B , Bernhard PJ . A local search optimization algorithm based on natural principles of gravitation. In: Proceedings of the 2003 interna- tional conference on information and knowledge engineering (IKE'03); 2003. p. 255–61 .
12. Erol OK , Eksin I . A new optimization method: big bang–big crunch. Adv Eng Softw 2006;37:106–11 .
13. Rashedi E , Nezamabadi-Pour H , Saryazdi S . GSA: a gravitational search algo- rithm. Inf Sci 2009;179:2232–48 .
14. Kaveh A , Talatahari S . A novel heuristic optimization method: charged system search. Acta Mech 2010;213:267–89 .

15. Formato RA . Central force optimization: A new metaheuristic with applica- tions in applied electromagnetics. Prog Electromag Res 2007;77:425–91 .