IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-8, Aug 2018 E-ISSN: 2347-2693

Voice and Data Packets Optimization using AI Algorithms in User Datagram Protocol

[1*]Ondiek J. Otieno, [2]Sibiyia L. Ongachi

[1]Jomo Kenyatta University of Agriculture and Technology, Department of Computing and Information Technology, P.O Box 6200, Nairobi, 00200, Kenya
[2]Egerton University, Department of Computer Science, P.O Box 536, Egerton, 20115, Kenya

**Corresponding Author: japhondiekh@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 14/Aug/2018, Published: 31/Aug/2018

Abstract- Video and data is becoming the dominant traffic over the Internet. To be able to detect, fix and replace lost data and voice packets along User Datagram Protocol (UDP), we apply artificial intelligent optimization The two artificial intelligence (AI) algorithms applied in this work are Modified Artificial Bee Colony (MABC) and Modified Particle Swarm Optimization (MPSO). These algorithms show great improvement in preventing packet loss and making transmission reliable. Further test of the MABC and MPSO reveals that there is improved optimization of data and voice packet delivery. Another comparison demonstrates that MPSO is significantly better than MABC in overall performance.

Keywords- Artificial Intelligence, Algorithms, Optimization, Artificial Bee Colony, Particle Swarm Optimization, UDP.

I. INTRODUCTION

Communications in industries have taken a leading role in voice and data applications at all levels. With the advent of improved technologies like INTERNET and fibre optics, industries must confront the reality of converging their data and voice applications to increase enterprise profits. The increased use of real-time voice and data in world-wide enterprises and business industrial has made it worthwhile and prudence for communication networks to provide reliable and flexible data transfers. While recognizing this fact, voice and data transmission has remained unreliable in the UDP protocol making it an emerging ground for continued dwindling of profits occasioned by data loss. This paper therefore is based on formulating and optimizing the transfer of voice and data in UDP protocol.

Due to increasing competition in the data and voice applications and the requirement of quality transmission, adequate optimization methods by use of AI, self-learning algorithms is becoming virtually important to both companies and consumers in data driven sectors. One of the strategies is to ensure low running costs for data and voice application by avoidance of any redundancy and loss. This paper therefore tries to design an innovative approach of embedding artificial reasoning at the UDP protocol. In many different sectors, optimization mechanisms have the capability to boost the performance of data and voice transfer in different applications and can equally be useful in user datagram protocols. Further, optimization algorithms have become the modest and efficient means of determining process in complex scenarios while employing intelligent analysis and decision making. The approach should be able to recover, detect, fix and replace the data/voice to the service without interruption. Existing problems of UDP are expressed below;

i) Packets droppings along transmission in UDP.
ii) UDP exposes any unreliability, connectionless of the underlying network protocol to the user's program
iii) Internet Protocol over unreliable media, where there is no guarantee of delivery, ordering or duplicate protection of packets.
iv) UDP does not acknowledge, order packets nor provide feedback to control rate of information flow.

II. OBJECTIVES

The goal of this paper is to optimize voice and data Packets using AI algorithms in User Datagram Protocol and describe the theory of Artificial Intelligence Optimization in UDP protocol by focusing on the following objectives:

i) Design AI optimizationalgorithm to optimize voice and data in UDP,
ii) Implement AIalgorithm to optimise voice and data in UDP and,
iii) Test and validate the Al algorithm in UDP.

III. CONTRIBUTIONS OF THIS RESEARCH TO THE SCIENTIFIC KNOWLEDGE

Overtime, other researchers have based their best arguments in the alternative transmission of data and voice packets within Transmission Control Protocol (TCP) which ensures that there is acknowledgment of transmitted data, error detection and ordered delivery. Less focus has been attributed to UDP which is considered light weight for its inability to be trusted with data, however it usage has remain in place for fast real time voice and packets delivery.

Therefore, the contribution of this research is to enhance trust in the transmission protocol of User Datagram by ensuring retransmission of packets whilst ensuring efficiency and avoiding dropping of packets which is the major undoing and ordered delivery of voice and data packets. As a result, the two applied AI algorithms (Artificial Bee Colony and Particle Swarm Optimization) significantly shows improved performance of voice and data packets transmission.

IV. RELATED WORK

From the analysis of the algorithms which other researchers have carried, focus is based on enhancing and preventing packet loss during the transmission. *Comer, (2006*) argues that the main reasons for packet loss in UDP is simple transmission that UDP uses where there is no protocol mechanism with no handshaking dialogues creating a scenario for overflow with incoming packets. This paper implements optimized packet loss prevention mechanism by use of Artificial Bee Colony algorithm and Particle Optimization based algorithms methods that will take responsibility for the lost packets. (*Comer, 2006*) further identifies that non-existence of handshaking mechanisms is the reason behind in-order delivery of packets [1].Particle Swarm Optimization (PSO) based searches also reduce the searching space and enhances quality of image and compression in video transmission [12].

UDP being a connectionless protocol, the capability for sequence, flow control and error detection is not provided; neither does it have any connection state maintenance requirement. Connectionless means no effort is made to setup a dedicated end to end connection [2]. UDP only sends small packets that can be held, or sends small bits without providing connection [3], [8]. Creating a relationship between voice and data packets is as important as making sure that data is transmitted safely to the end.

V. METHODOLGY

This paper attempts to develop optimized algorithm from Particle Swarm optimization and Artificial Bee Colony optimization. Artificial Bee Colony based algorithm research combinational intelligence which was originally conducted by Dorigo et al. in the early 1990s for application in field of robotics and optimization [4].

The algorithms are optimized in the User Datagram Protocol to hence the delivery of voice and data packets to prevent loss and non-delivery. The methodology carries an iterative process of transmission of voice operation, and data packets. Using algorithm design approach, the system of UDP reliability, flow and error control and queuing are tested. A general workflow that involves the following phases are modelled as follows;

i) Analysis,
ii) Design,
iii) Implementation and,
iv) Testing

Although the design majors in two AI algorithms: The Artificial Bee Colony and Particle Swarm Optimization development process were improved to accommodate the use of UDP protocol operations structure in voice and data transmission.

A. Optimization.

Optimization problems are those disparities that in one way or another exhibits solution which yield highest or lowest performance [5]. AI swarm optimization methods: Artificial Bee Colony Optimization and Particle swarm optimization which are promising techniques in evolutionary computation were majored on to achieve results. Optimization being the process of finding the best results under maximum or minimum parameters, ABC and PSO have significantly shown good progress in the research was based on techniques from evolutionary computing of subfield that arose from metaheuristic and heuristic nature.

VI. DESIGN AND IMPLEMENTATION

A. UDP Design.

The UDP design for voice and data transmission is described in figure 6.2. This structure highlights the general transmission of voice and data packets in the UDP layer which has been derived from the TCP/IP protocol suite described in the figure 6.1 below. The application of ABC and PSO algorithm will therefore be used at different stages of problems sources stated. The specific problems to be implemented in optimized ABC and PSO intelligence are:

1. Preventing the dropping of Voice and Data Packets;
2. Creating reliability in the transmission process using ACO/PSO optimization algorithm by solving queuing problems of voice and data packets at the control block module through: Identification of the packets sizes and ports that are easy to reach;
3. Flow and Error control by managing the flow of voice and data packets using pheromone technique not to allow dropping of the packets, if not transmitted, it is transmitted by the virtue of its infection by pheromone;

4. No queuing, identifying the fastest possible route available for routing transmission even for larger packets.

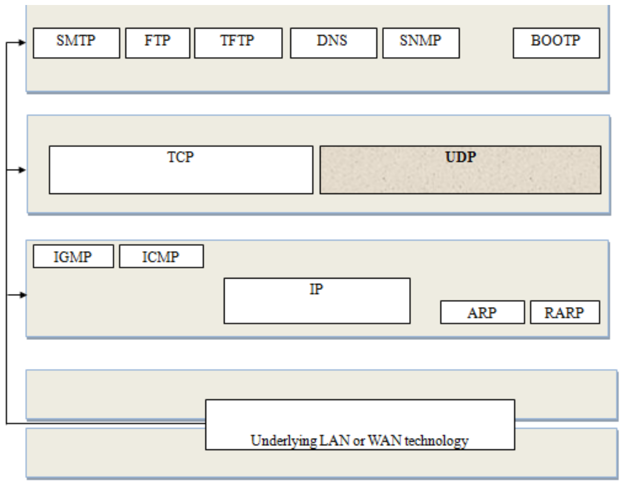


Figure 6.1: UDP in the TCP/IP protocol suite: Source (Foruzan, B 2003).

B. UDP Voice and Data packet design.

Input Model and output model are structured such that checksum algorithms employs pseudo header in the checksum calculations to verify and optimize the delivery of the message such that when message is changed while on transit to cause misdelivery, the UDP checksum would detect this and drop the packets before reaching the ports though this is optional in UDP. Control block table has well known ports for UDP including the two identifiers, the IP address and the port number at each end to make a connection.

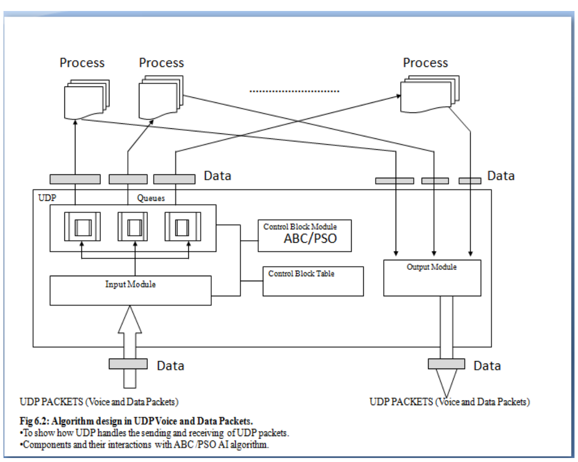


Figure 4.0: UDP design for voice and data transmission

C. Modified PSO Algorithm Implementation.

The PSO is a population based optimization technique based on swarms, [6]. In this implementation, design and analysis, the swarm will be represented by the data and voice Packets with the ability to use information to make decision and move in search space for destination, however memorizing previous position will not be applicable as the voice and data packets are one-time transmission unless the destination port is not located. Along transmission, fitness of a particle is computed as the sum of absolute difference between expected output value and value of transmitted packet.

$$\text{Packet fitness} = \sum [ExpectedValue - ComputedValue]$$

The best packets for transmission are the ones that output better approximations of the expected values where packet A is better than particle B if fitness of A is lower than another of B. The packets swarm consists of N packets with the intention to be transmitted around to the destination in D-dimension port search space. In this research improvement, every packet makes use of its own memory and knowledge gained by the packets as a whole to find the best solution.

The value pbest is introduced as the best packet previously visited position of the i_{th} packet denoted by; $Pb_i = [Pb_{i1}, Pb_{i2} \ldots\ldots Pb_{iD}]$.The gbest value is the global best position of the all individual pbest values denoted by; $g_{bi} = [gb_1, gb_2, \ldots\ldots gb_d]$.The position of the i_{th} Packet is denoted by $X_i = X_{id}^{new} [X_{i1} \ldots\ldots, X_{i2}, \ldots\ldots X_{iD}]$ .The position and speed of the packets are updated by $pbest_i$ and gbest in each of the transmission. This uprise to updated equation which can be formulated below;

$$V_{id}^{new} = W \times V_{id}^{old} + C_1 \times r_1 \times \left(Pbest_{id} - X_{id}^{old}\right) + C_2 \times r_2 \times \left(gbest_d - X_{id}^{old}\right) \ldots \text{Equation 3.}$$

$$X_{id}^{new} = X_{id}^{old} + V_{id}^{new}$$

Where r_1 and r_2 are random numbers between (0, 1), C1 and C_2 will control how a packet in their generation interact socially constants.

V_{id}^{new} and V_{id}^{old} denotes speed of both new and old packets respectively while X_{id}^{old} denotes the current packet position at iteration K; X_{id}^{new} remains the updated packet position. The packet weight W controls the impact of the earlier previous speed of packet on its current one; W will be designed to replace V_{max} and adjust the influence of earlier packet speeds on the optimization process.

$$W^k = W^{max} - \frac{W^{max} - W^{min}}{Iteration^{Max}} * Iteration \ldots \text{Eq 5.}$$

W^k=Packet size factor providing flexibility to control the balance between small and large packets sizes where;

W^{max}=Initial packet size.
W^{min}=Final packet size
Maximum iteration=Maximum iteration No.
Iteration=Current iteration No=1.

The modified steps of the algorithm undertake to achieve the optimized transmission of data and voice packets is iterated in figure 6.3 below and follows the following pseudocode;

Step 1: Set the packets dimensions and sizes in the queue as all equal and important.

Step 2: Initialize the packets randomly at source W^{max} determining size. From $i_{source} \dots \dots \dots j_{port}$ and time $C_{ij}.X_{ij}$

Step 3: Compute size values; this step majorly addresses starting point of iteration. Before packet transmission, position source and destinations are updated based on the packet size considering W^{max}=Initial packet size and W^{min}=Final packet size. The speed to be used velocity, $V_i = [V_{i1}, Vi2 \dots V_{iD}]$.

$$= \sum_{j=1}^{m} X_{ij} \sum_{i=1}^{n} X_{ij}$$

Step 4: For each packet, calculate it fitness value as provided in the equation.

$$\text{Packet fitness} = \sum [ExpectedValue - ComputedValue]$$

Step 5: Find the best packet size and corresponding best destination port. The choice is based on many different motivations including packets size; W^{max}=Initial packet size.

W^{min}=Final packet size, destination port, error free data.

Step 6: If fitness value is better that the previous best *pbest*, set the current local transmission fitness values as the new *pbest*.

$$X_{id}^{new} = X_{id}^{old} + V_{id}^{new}$$

In this context, transmission is what is regarded as *fitness value* and not the packet.

Step 7: Select the best route over which transmission took place based on destinations as *gbest.*

Step 8: For all packets, update using equation 1;

$$V_{id}^{new} = W \times V_{id}^{old} + C_1 \times r_1 \times (Pbest_{id} - X_{id}^{old}) + C_2 \times r_2 \times (gbest_d -$$

Step 9: Repeat the process until maximum iterations is satisfied and all packets transferred, if not then repeat from step 3. Ideally no packet droppings are expected since transmission is based on finding the best fit routes, transmission time and destination processes.

In PSO, a large number of real life applications can be formulated as a voice and data transmissions. It seeks to optimize flow; dropping of packets and error control that ideally originates from a number of sources and are destined to number of destinations. Therefore, given there are n sources and m destinations, the size of data at source ***i*** is $\boldsymbol{a_i}$ and the port destinations ***j*** is $\boldsymbol{b_j}$.

The unit time transportation for packets between ***i*** and destination ***j*** is $\boldsymbol{C_{ij}.X_{ij}}$ which is the time from source *i* to the destination ***j*** creating the mathematical model of;

$$= \sum_{i=j}^{n} \sum_{j=1}^{m} C_{ij} X_{ij} \dots\dots\dots \text{Equation. 2}$$

The port source and port destination is denoted below; this denotation helps in determining the congestion of the incoming packets, eliminates queuing by adjusting transmission rate and speed hence optimizing flow control mechanism using;

Destination Voice and Data Packets:

$$= \sum_{j=1}^{m} X_{ij} \leq a_i \text{ [i=1, n] ....Equation.3 (a)}$$

Source Voice and Data Packets:

$$= \sum_{i=1}^{n} X_{ij} \geq b_j \text{ [j=1, 2,......m]...Equation.3 (b)}$$

The initial determination of packet size; source and port Number would help optimize a number of variables depending on the time of transmission $\boldsymbol{C_{ij}. X_i}$ this helps in size overflow management, undue queuing hence absolving packets from imminent loss through droppings.

D. Flow and Error Control in PSO.

PSO creates ability for flow control by allowing incoming data packets flow without silently discarding voice and data packets. After each queue process for both incoming and outgoing packets associated with each process terminates the best routes, best time benchmarks, port destinations are stored for the next transmission reference.

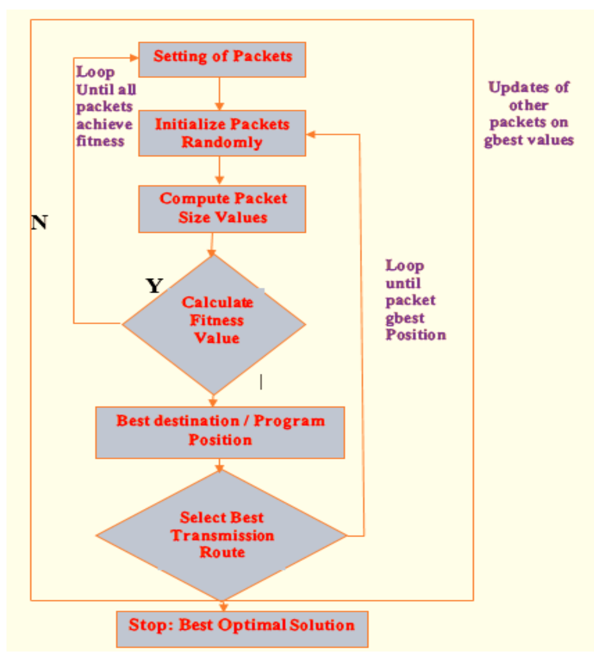


Figure 6.3: Flow Diagram of Modified PSO Algorithm

E. Modified artificial bee colony (ABC) Algorithm Implementation.

In ABC, the honey bee colony acts as the search for new sources of food around their hives; leading to the composition of the algorithm. Employing the colony to consist of three bees in groups as scouts, on looker bees and employed bees. This is summed up in employing half of the artificial bees and the other half as food sources making the number of food sources equal to the employed bee number around the hive [7].

In this context, queuing problems arise in situations where there are more incoming voice and data packets. Let's consider a scenario where there are x, x={1,2,3,4……..x}packets to be transmitted and there are y, y={1,2,3,4,5 ….. y} ports destination available with the condition that packets are not permitted to be swapped and migrated between ports. In such a case, if we have y>x, then at this point, there would be no requirements for developing algorithms in UDP since voice and data packets will be allocated the port on the first come scenario. But if y<x, then developing new optimized algorithm for packet allocation would be ideal since there would be inefficient transmission, queuing which deter throughput, dropping of packets and hampering of efficiency.

In this implementation, the cycle of the search will be addressed in iterative steps consisting of;

i) Moving onlooker bees and employed bees onto the food sources (Port destination)
ii) Calculating the food nectar amounts (destination packet size) and
iii) Directing the determined scout bees onto possible food sources.

The food source position will be the ideal solution of the problem to be optimized representing the best route and port for transmission while the amount of nectar in the food source will represent the quality (packet size) of the solution represented. Onlooker bees on the food sources with selection process of probability are selected. When nectar food sources (port destinations) amount increases, the value of probability with which food sources is liked by onlooker bees (Voice and data packets) increases as well. In every bee colony, there are scouts which are colony's explorers-they don't have guidance while looking for food and are always concerned with finding any available food source.

At the first step, ABC generates incoming packets randomly P(C=O) of SN solutions (food source positions), where SN denotes Size of Packets to be transmitted (in this case onlooker-or employed bees). Each solution X_i (i=1, 2, …….SN) is a D-dimensional vector. Once the initiation is done, the packet positions solution is subjected to repeated cycles equivalent to C=1,2,3……….MCN of the search process of the packets to be transmitted and the scout bees (Packets on queue).In this model, C=1…..,MCN is preferred to avoid packet delay and search which will prevent achieving the intentions of optimization. The first transmitted packet produces a modification on the position solution (Packet port destination); fastest route in packet memory depending on local transmission information and will test the port destinations (fitness values) on the new source (new solution).Should the port destination be favorably faster and free of error; then the packets memorize the new port position and forget the old(drops old port) one or else it keeps the previous position of optimum transmission meaning it has achieved the optimal values.

F. The Modified ABC algorithm Pseudo code implementation.

As shown in the figure 6.4, an iterative proposed flow of the modified ABC algorithm is presented in pseudo code below;

1. Initialize the population of solution
   X_i, i=1…….SN, j=1…….D
2. Evaluate the population of voice and data packets
3. Cycle=1 (creating 1 cycle to improve optimization)
4. Produce new solutions V_i for the (employed) bees using the evaluation $V_{ij}=\varphi_{ij}\ (X_{ij}-X_{kj})$ and evaluate.
5. Calculate the probability value P_i for the solutions X_i, using calculation;
   $$P_i = \frac{fit_1}{\sum_{n-1}^{SN} fit_n}$$
6. Depending on the new solution V_i for onlooker bee from the solutions X_i selected depending on P_i and evaluate them.
7. Using the evaluation, replace the Scout based on random solution depending on the produced solution X_{ij}.
8. Memorize the best solution achieved so far.
9. Replicate the cycle until MCN for all Packets Cycle+1 until cycle=MCN.

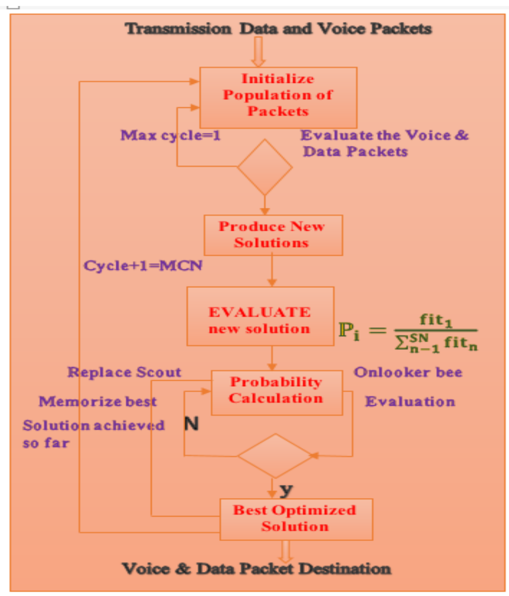


Figure 6.4: Flow Diagram of modified ABC Algorithm

G. Assumptions in Algorithms Design.

i) No relationship exists between the user datagrams even if they are coming from the same sources and going to the same destination port.

ii) Voice and data packets are not numbered and there is there is no connection establishment and termination - This is useful in that packets can travel on different paths.

iii) Each request must be small enough to fit into one user datagram and therefore there is limit for only short messages and voice streams that can be sent through UDP [8],[9].

iv) No acknowledgement of transferred data from destination to source in MABC and MPSO.

v) During transmission, the eventual delay can be realized but data will never be dropped as there is opportunity for re transmission.

VII. EVALUATION AND TESTING

The modified proposed algorithms will present good results for specific types of problems that are realized in the transmission of voice and data. Classical benchmark problems are frequently used to examine the performance of a proposed algorithm or a proposed modification on the existing algorithms [10]. An existing UDPTEST Tool 3.0 is used to analyze voice and data packets

A. Tools for the Experiment.

i. The experiments are performed on the same computer with Intel (R) Duo Core (TM) 2 CPU T6400 (2.00 GHz) and 2 GB RAM. The algorithms are embedded in ready platforms on andre-matlab-v0.0.2 that uses C++ coding benchmarking optimization functions and compiled under the Windows 7.

ii. Algorithms were started within optimum cycles; the function calls limit was set to 10000kb because of insufficient memory).

iii. MPSO and MABC Test were performed on andre-matlab-v0.0.2 with Step ***f1*** and Sphere ***f2*** optimization functions.

Step:

$$f_{2}=\sum_{t=1}^{D}(|X_1+0.5|)^2\ldots\ldots\ldots\ldots\ldots\ldots.[-100,100]$$

Sphere:

$$f_1=\sum_{t=1}^{D}X_1^2\ldots\ldots\ldots.[-100,100]$$

B. Experimental Results.

A function is multimodal if it has two or more local optima. A function of variables can be separable if it can be rewritten as a sum of functions of just one [11]. The separability is closely related to the concept of interrelation among the variables of the function. Optimization techniques are evaluated by comparing them to other optimization techniques in terms of the number of function evaluations required to find a given optimum or the best objective value found [13].

The results are based on the test carried on both normal UDP TEST software and the two Modified artificial intelligence optimization tools, MABC and MPSO respectively for comparison and further analysis.

Table 5.2.1**:** UDP TEST tool results for sent voice and data

UDP TEST TOOL				
Functions	Sent Packets	Mean of Received Packets	Mean Time(s)	Dropped Packets
F1-Test 1(10 runs)	7896kb	7754kb	183	142
F2-Test 2(10 runs)	9867kb	9787kb	194	80
F3-Test 3(10 runs)	4578kb	4366kb	147	212
F4-Test 4(10 runs)	6800kb	6793kb	168	7
F5-Test 5(10 Runs)	9523kb	9522kb	172	1

packets

UDP Test Tool Sent Packets

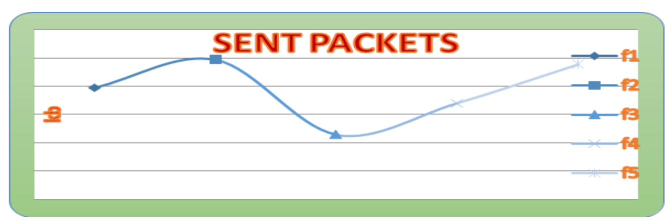


Figure 6.5: Size of Sent Packets in UDP

Comparison of sent according to figure 6.5 and received in figure 6.6 data indicates variation since most voice and data are lost during transmission.

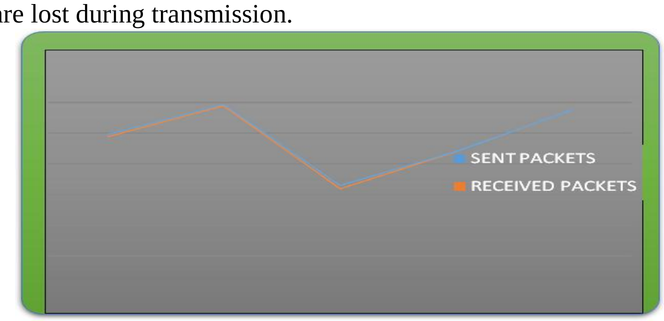


Figure 6.6: Comparison of Sent and Received Packets in UDP

From the figure 6.6, it can be noted that received data is less than sent data due to voice and data packets losses during transmission.

Figure 6.7 compares different test runs of sent and received packets against time. Different test runs occupy different time depending on the condition and environment of tests.

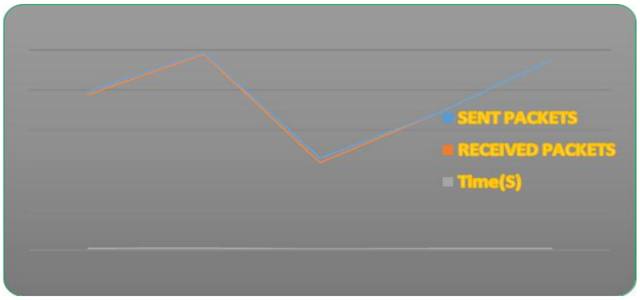


Figure 6.7: Time(s) against sent and received voice and data packets.

C. *Comparative analysis of the UDPTEST TOOL Software 3.0, MPSO and MABC.*

The first two function measures with the same sent packets is used to analyzed the findings of received and dropped packets. The tests are subjected to same environmental tests.

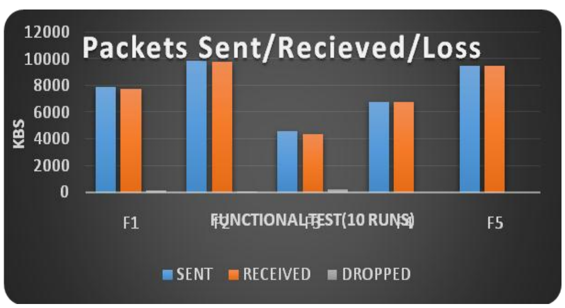


Figure 6.8: Graph analysis of Sent data, Received Packets and Packets Loss

In the run tests for MPSO and MABC Algorithms the variables for determining the optimized voice and data packets transfers will be addressed using benchmarked optimization functions from Step ***f1*** and Sphere ***f2*** optimization functions. The pseudo codes implemented are automatically generated by the andre-matlab-v0.0.2 software.

D. *MABC Algorithm test.*

ABC algorithm has few control parameters: Maximum Number of Cycles (MCN). Each of the experiments were repeated 10 times in a repetitive same packets size to get the mean of received packets and dropped packets as shown in the table 5.4.1 below.

Table 5.4.1: MABC Algorithm Test Results

MABC Algorithm Test Results					
Optimizations Functions	Sent Packets	Cycles	Time (s)	Mean Received Packets	Dropped Packets (kb)
F1 (Step)	7896kb	125	78	7890kb	6kb
F2 (Sphere)	9867kb	150	97	9859kb	8kb

From the figure 6.9 and figure 6.8, it can be analyzed that function F1 (Step) maintains more less the same data sent and received. This is due to the optimality function provided by the test runs in MABC. Minimal voice and data packets loss is recorded. The test functionality for F2 (sphere) also applies the same parameter

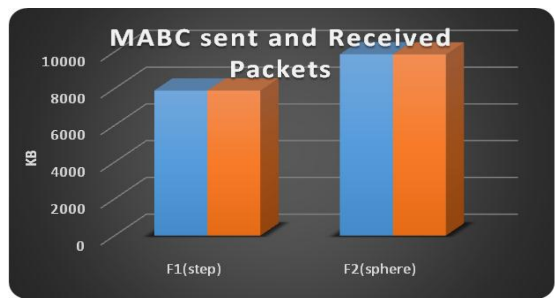


Figure 6.8: Sent and received Packets in MABC test runs using F1 and F2 functions

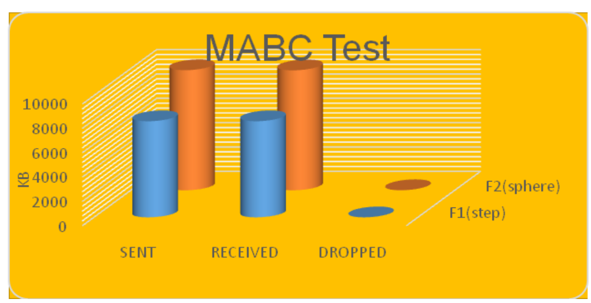


Figure 6.9**:** Sent, received and lost Packets in MABC test runs using F1 and F2 functions

E. *MPSO Algorithm Tests*

Each of the experiments were repeated 10 times in a repetitive same packets size to get the mean of received packets and dropped packets for the functions F1 (Step) and F2 (Sphere) as shown in the table 5.4.2 below.

Table 5.4.2**:** MPSO Algorithm test results

MPSO Algorithm Tests					
	Sent Packets	Cycles	Time (s)	Mean Received Packets	Dropped Packets (kb)
1	7896kb	114	45	7895kb	1kb
2	9867kb	128	42	9854kb	3kb

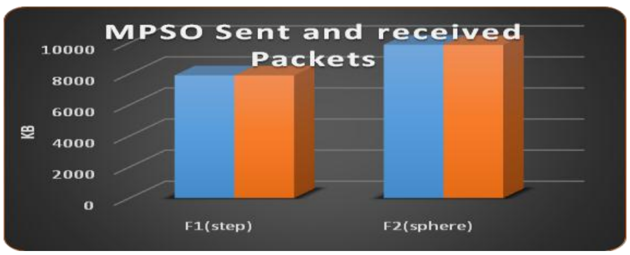


Figure 7.0: MPSO Sent and Received test results in F1 and F2 functions

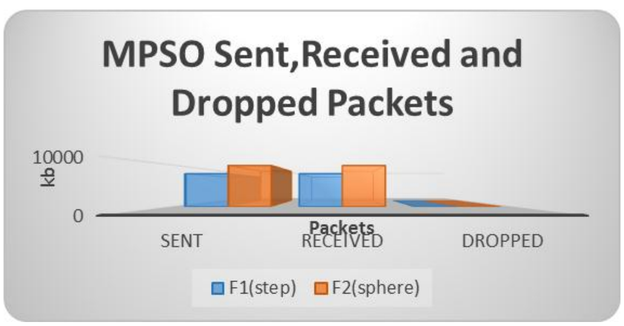


Figure 7.1: MPSO Sent, Received and dropped packets using F1 and F2 functions.

From the figure 7.1, it can be concluded that both F1 (Step) and F2 (Sphere) record faster rate of transmission and minimal drop of voice and data packets.

F. Comparison of MPSO and MABC.

	MPSO				MABC			
	Sent (kb)	**Recei ved**	**Time (s)**	**Drop ped**	**Sent(kb)**	**Recei ved**	**Time (s)**	**Drop ped**
1	7896	7895	45	1	7896	7895	78	6
2	**9867**	**9854**	**42**	**3**	**9867**	**9859**	**97**	**8**

Comparative analysis of Modified Artificial Bee Colony and Modified Particle Swarm Optimisation is described and tabulated in the table 5.6.1 below.

Table 5.6.1: Comparative analysis for MPSO and MABC

G. Statistical Analysis of Results.

The proposed modified algorithms were run on an array of common and existing benchmarks for 10 evaluations for per function. These two benchmarks were chosen for their variety functions of simple unimodal problems. Performance was measured as the minimum error was measured as the minimum error $|f(x) - f(x\_)|$ found over the trial, where $f(x\_)$ is the optimum fitness for the problem. Results were averaged over five independent trials (for UDPTest Tool), and are displayed in Table 5.6.1.

A number of tests runs are conducted and values recorded. Values are then ranked according to function types. The researcher then goes down the list to check consistency of sent and received data; this procedure validates tests which will be used to compare the two algorithm groups

After gathering empirical data, difference in performance between MABC and MPSO algorithms were notable between the two benchmark functions. One version was faster and optimal after a number of functional evaluation and iterations which were averaged after a number of trials. The tests were based on pairs of groups of data to determine whether a difference is significant or not. Packet loss in UDP is caused by congestion leading to several queues along the port, the tested AI optimization algorithms brings the strengths of faster cycles and less time for delivery, this prevents queuing that is the main caused for data loss hence greater control of error. Even though the main aim of the algorithm was to eliminate drooping of packets, it still remained significantly lower in both MABC and MPSO.

The results of this AI algorithm displays consistency in delivery of voice and data packets without change a long transmission unlike checksum algorithm which employs pseudo header in the checksum to verify and optimize the delivery of the message such that when message can be changed while on transit to cause misdelivery.

Using Data Analysis, it’s evident that the modified AI optimization techniques out do the UDP streaming and transmission of packets. The algorithms performance in limited seconds and depicting of minimal drop of packets proved that optimization could be achieved using the artificial intelligence methods. The sum up is that reliability and error control, queuing delays and flow control are improved compared to the criticized UDP data transmission model.

VIII. CONCLUSION

PSO has received considerable number of research modifications and variations of the original algorithm creating improvements on the original. Artificial Bee Colony is also a new optimization technique which has shown to be competitive to other population-based stochastic algorithms. However, from our results Artificial Bee Colony and other stochastic algorithms suffer from the same problems arising from the convergence speed of ABC which is typically slower than PSO.

Further complex analysis reveals that ABC algorithm easily gets stuck when handling complex multimodal problems and doesn't capture all variables. From the search pattern of both employed and onlooker bees, there is good exploration however exploitation is very poor. In order to balance the exploration and exploitation of ABC, this paper proposes a new MABC variant (MABC). This approach slightly differs from other improved ABC algorithms. In ratifying the performance, a set of two benchmark functions are used in the experiments. Comparison of MABC with normal ABC demonstrates this new search pattern can effectively accelerate the convergence speed and improve the accuracy of packet transmission solutions. Another comparison demonstrates that MPSO is significantly better than MABC. The approach is easier to implement and simpler.

Owing to rigorous test runs carried across the tools and the overall development of the algorithms, several conclusions can be considered;

i) Performances of Modified Artificial Bee Colony (MABC) with that of Modified Particle Swarm Optimization (MPSO) were tested on two high dimensional numerical benchmark functions that have multimodality.

ii) From the results simulated it was concluded that the proposed algorithm has the ability to optimize data and voice packets in terms of reducing packet drops and error control, flow control, ordering and improving time efficiency.

iii) The results didn't portray any feedback mechanisms due to incapability of the test software and the limitation realized in developing the algorithm.

iv) Statistical summary according to achieved results in table 5.6.1-comparative analysis of MPSO and MABC.MPSO in f1 sent 7896kb and received 7895kb within 45s and dropping 1packet while MABC sent 7895kb and received 7895kb in 78s and dropped 6 packets. In f2 MPSO sent 9867kb receiving 9854kb in 42 s and dropped 3packets while MABC sent 9867kb receiving 9859 at 97s dropping 8packets.From analysis MPSO performs slightly better than MABC.

Recommendation

There are several issues which remain as the scopes for future studies such as the investigation of the control parameters' effect on the performance of the ABC and PSO algorithm and the convergence speed of the algorithms and the unprecedented drop of packets. The scope of variables pick up, cycle time should be researched on to provide better testing environments for better results.

REFERENCES

[1] Douglas E. Comer 2006, Internetworking with Transport Control Protocol /IP, Protocols, Principles and Architectures', Vol.1, 5th Edition.

[2] Yashpaul ER., Singh and Swarup A. 2009, "Analysing Adaptive Routing Algorithms using Connectionless Protocols", *World Academy of, Engineering Sciences & Technology* No.28.

[3] Behrouz A. 2003, 'Data Networks and Communications', TATA MCGRAW-HILL 3rd Edition

[4] Dorigo M, Bonabeau E, G. Theraulaz 1999, "Artificial *Swarm Intelligence from Natural to Artificial System"*, Oxford University Press

[5] Wong, K. C. P.; Ryan, H. M. and Tindle, J 1996. Early warning fault detection using artificial intelligent methods. 31st Universities Power Engineering Conference18th -20th September 2006, Crete, Greece.

[6] Li-Yeh Chuang, Yu-Da Lin, and C.-H. Yang 2012, "An Improved Particle Swarm Optimization for Data Clustering," in Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2012, IMECS 2012 Hong Kong, 2012, pp.440-445.

[7] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Erciyes University, Engineering Faculty, Computer Engineering Department, Turkey, Technical Report-TR06, 2005.

[8] Stallings, W 2007, 'Handbook of computer-communication standards', Volume 2: *Local Network*. Macmillan Publishers, New York.

[9] Dorigo M, Mauro B, and Thomas S., 2006 IEEE Computational Magazine on Intelligence e Librederuxeles, Belgium November.

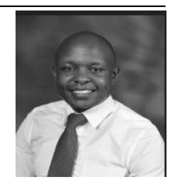
[10] Yao X., Liu, Y. & Lin, G. 1999. Evolutionary Programming Made Faster.IEEE Transactions On Evolutionary Computation, VOL. 3, NO. 2.

[11] Hadley, G.1964: Nonlinear and Dynamics Programming. Addison Wesley, Reading, MA.

[12] Shraddha Pandit et.al, 2008. Enhance the Performance of Video Compression Based on Fractal H-V Partition Technique with Particle Swarm Optimizatio International Journal of Computer Sciences and Engineering Vol.6(1), Jan 2018, E-ISSN: 2347-2693

[13] Wolpert D.H and Macready W.G. 1997. "Theorem for optimization and Evolutionary Applied Computation", volume1, page 45.

Authors Profile:



Japheth Ondiek is currently pursuing PhD Computer Science at the Jomo Kenyatta university of Agricultural Technology (JKUAT). He has published several papers in Machine Learning for development and has also authored several pieces for Human centered computing and design thinking for innovation. His research interest includes Artificial Intelligence, Knowledge Based Systems, Machine Learning, Human Centered Design.

Lynet Sibiyia pursed BSc in Applied Computer Science (Information Technology option) from Africa Virtual University – RMIT (Royal Melbourn Institute of Technolgy, Australia), hosted by Egerton University, Nakuru Campus Kenya, in 2008 and Master of Science in Data Communication from KCA University, Kenya in year 2014. She is currently working as Senior Technologist in the Department of Computer Science Egerton University. She has published research paper in International Journal of Computer Trends and Technology (IJCTT). She has 6 years of teaching experience and 3 years of Research Experience.

