Business, Management and Economics Engineering
ISSN: 2669-2481 / eISSN: 2669-249X
2023 Volume 21 Issue 02

IMPROVING RELIABILITY IN UBIQUITOUS COMPUTING SYSTEMS WITH SOCIAL SPIDER OPTIMIZATION ALGORITHM

[1]Azhar Ameer Hamza , [2]Seyed Ebrahim Dashti (corresponding author), [3] Hala Hussein Issa Allobawi

[1, 3] Department of Computer Engineering, Shiraz Branch, Islamic Azad University

[2] Department of Computer Engineering, Jahrom Branch, Islamic Azad University

ABSTRACT

Ubiquitous computing refers to the distribution of data over a large amount of data sources. Such a computing system gathers millions of resources to enable parallel computing applications Its goal is to harness the Internet's vast computational capacity for large parallel applications. Since many users deal with these systems day and night, reliability and throughput are very important to them, therefore, this paper focuses on improving them in ubiquitous computing systems. This article suggests a spider optimization algorithm to improve parameters like reliability and throughput. It compares with other evolutionary algorithms; the result of the simulation shows significant improvement in comparison to other algorithms.

Keywords: Ubiquitous Computing, Network Computing, Social Spider Optimization Algorithm, reliability, throughput.

1-INTRODUCTION

Today, ubiquitous computing (global computing) has been introduced to share the power of all parallel computers to provide resources to users in the most remote locations. Ubiquitous computing is a new concept that includes a large collection of computers connected with the help of the Internet to be able to use their processing power to work on time-consuming processes [1]. Ubiquitous computing has the idea of harnessing the computational power of networked computers. It allows users to access and share computational resources including hardware, software, and data published by others across a wide area network (WAN) [2]. Ubiquitous computing means collecting the idle time of computers connected to the Internet and using them to solve problems on a massive scale. Therefore, ubiquitous computing can extend the cycle theft model throughout the Internet. A typical architecture of ubiquitous computing systems is shown in Figure 1 [3]. The ubiquitous computing system may be regarded as the natural evolution of the Internet. It can harness ubiquitous resources (CPU time, memory, disk storage) and make them available to any user on the Internet. In such a ubiquitous computing environment, mobile devices contain various kinds of data: content data, profile data, and essential metadata [4].

The main components of a ubiquitous computing system include the following [5]:

- Sharing by several clients: Computing servers can be shared by several clients in the network, where the computing task is shared.

- Communication performance: Ubiquitous computing requires large amounts of data to be sent over the network.

- Remote Library Design and Reuse: When executing remote libraries on the computing servers, there is an option as to whether to distribute the computing resources amongst different client requests in a task parallel manner or to allocate all the processors to each client task.

The main feature of a global computer is its geographical distribution and its widespread use, because in ubiquitous computing it connects different devices such as clusters, PCs, PDAs, handsets, and mobile phones that may be widely dispersed geographically, and this makes the issue of reliability vital. [3].

In this article: Section 2 discusses relevant work that has already been done; Section 3 provides a review of ubiquitous computing. The social spider optimization algorithm was explained in Section 4andin Section 5, proposed new modeling of the variable through the web. In section 6, we have the Evaluation, and Finally, in section 7conclusion.

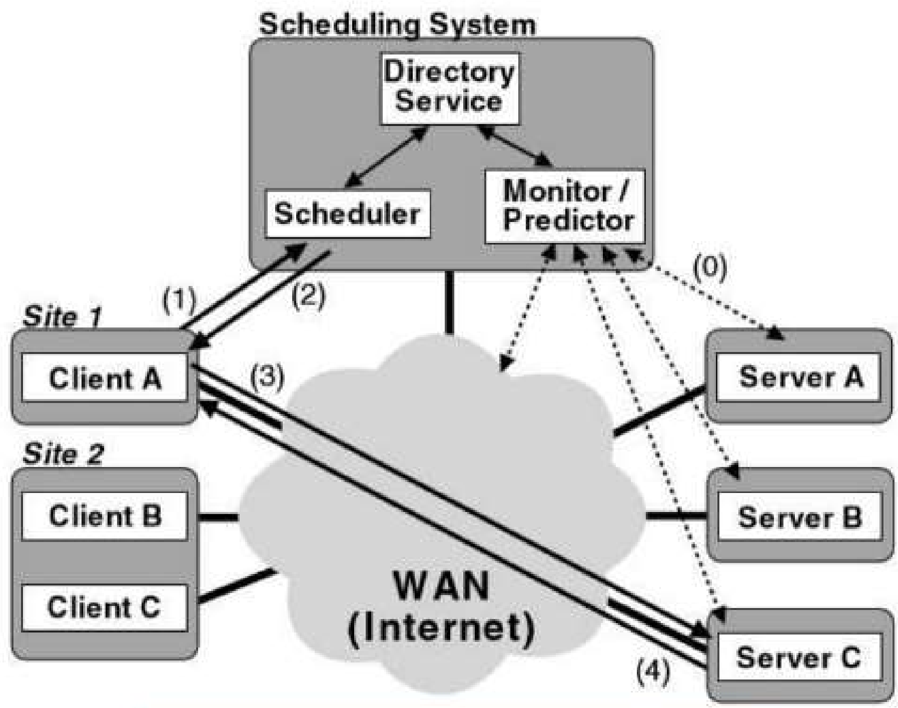

Fig-1: A typical architecture of ubiquitous computing systems [3].

2-EXISTING RELATED WORK

The scale of the Global Computing System (GCS) paradigm requires to review of the basic issues of distributed systems, i.e. security, performance models, scalability, and fault tolerance. Researchers in this paper review recent work in Ubiquitous Computing, in Peer-to-Peer systems. Also, present XtremWeb, the Ubiquitous Computing System we are currently developing[2].

In another paper, the researchers focused on ubiquitous computing, according to them ubiquitous computing represents a paradigm shift from storing data in monolithic data management systems towards seeing it distributed over a large number of mobile, small, data-carrying devices. In another paper, the researchers focused on ubiquitous computing, which they say represents a shift in the paradigm of data storage in integrated data management systems toward its distribution across a large number of small, mobile, and portable devices. The ubiquitous computing system can be known as the natural evolution of the Internet, which is related to network computing, cluster computing, and Internet computing[3,4].

Swarm intelligence mimics the nature methods to solve optimization problems. There are different metaheuristic algorithms used to drive a search for the optimal solution such as Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Firefly Algorithm (FA), Genetic Algorithm (GA), Artificial Bee Colony Optimization (ABC), Social Spider Algorithm (SSA) and Social Spider Optimization Algorithm (SSO).

In general, the success of evolutionary algorithms (EAs) in solving optimization problems relies on the proper selection of control parameters [6]. Control parameter tuning is the process where different combinations of parameters are assessed to achieve the best performance in terms of solution quality and/or computational complexity. Parameter sensitivity analysis methods and applications have attracted much research attention in the past decades and much work has been done to fully exploit the searching power of EAs [7]. An early example of such applications was conducted by Grefenstette, in which the control parameters of a genetic algorithm were tuned by another one to improve the performance of an image registration problem. So far, parameter sensitivity tests have been performed on most of the existing EAs, for example, PSO [9] and DE [10].

Social Spider Algorithm (SSA) is a recently proposed metaheuristic designed to solve real-parameter black-box optimization problems [11]. SSA performs optimization by mimicking the search behavior of social spiders [12]; The same process has been used in solving benchmark optimization problems [11] and real-world problems [13] [14] and has resulted in satisfactory performance. Although the performance of SSA in existing work outperforms the compared algorithms, it is influenced by the four control parameters (population size, attenuation rate, change rate, and mask rate). Therefore, the correct setting of the parameters is necessary to show the efficiency of SSA as a problem solver. SSO formulates the search space as a web on which each position represents a feasible solution to the optimization problem and all feasible solutions to the problem have corresponding positions on this web [15, 16]. Each spider on the web holds fitness value which is based upon the objective function, and represented by the potential of finding a food source at the position. SSA is used for solving constrained optimization problems[15, 17].

ABC depends upon the intelligent foraging behavior of bees. The number of working bees is equal to the number of food sources. This bees do waggle dance when they choose their food source. Onlooker bees watch the dance of employed bees and select one of their food sources depending upon the dance and move towards that, after that it evaluates its nectar amount. [18, 19].

3-UBIQUITOUS COMPUTING

Ubiquitous computing is beneficial for a user who is looking for some extra computational power to run parallel supercomputing applications[21]. An important feature of ubiquitous computing systems is location transparency; information can be obtained irrespective of time or location[21]. Corporations can use ubiquitous computers to run large parallel and distributed applications with performance far superior to today's supercomputers, at a fraction of the cost [21].

A major challenge will be how multiple, heterogeneous ubiquitous computers can interact effectively and securely. Although ubiquitous computations are expected to tolerate low rates of errors, massive attacks occur when the error rates are larger than tolerable. Such widespread attacks may occur due to distributed denial of service. With today's internet that does not have enough bandwidth, this becomes a bottleneck. [22].

4-SOCIAL SPIDER OPTIMIZATION ALGORITHM

A majority of the spiders are solitary which means that they spend most of their lives without interacting with others. Among the 35000 spider species observed and described by scientists, some species are social. These spiders live in groups. Based on these social spiders, social spider optimization (SSO) was developed to optimize the problems. There are two components of an SSColony, social members and communal web. The social members are divided into females and males. The number of female spiders reaches 70%, while the number of male spiders reaches 30% of the total colony members. Female spider presents an attraction or dislike to other spiders according to their vibrations based on the weight and distance of the members[23].

In the Social Spider Optimization (SSO) algorithm, the shared web represents the search space. The search space of the optimization problem is seen as a hyper-dimensional spider web. Each solution in the search space also represents the position of the spider. The weight of each spider represents the fitness value. The search behavior of spiders can also be explained as the cooperative movement of spiders toward the food source. Spiders are very sensitive to vibratory stimulation as vibrations on their webs notify them of the capture of prey. Each of the spiders on the web has a position and solution fitness value. When a spider is moving to a new position, it creates a vibration that propagates through the web. Each vibration holds the information of one spider and other spiders can get the information [24].

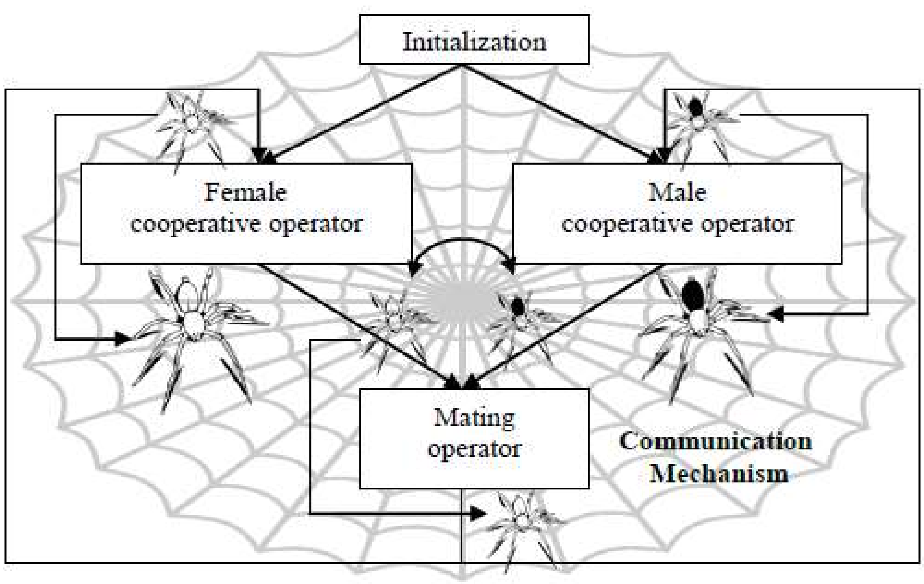

Fig-2: Schematic data-flow[23]

4-1- Differences between SSO and Other Evolutionary Computation Algorithms

Many meta-heuristic algorithms have been introduced in recent years. Among them, PSO and ACO are the most widely used algorithms studied.SSO may also be classified as a meta-processing algorithm, but it has many differences from PSO and ACO, which are explained below.

PSO, like SSO, was originally proposed to solve optimization problems inspired by animal behavior. However, the first important difference between SSO and PSO is in the following patterns. In PSO, all particles follow the joint best global position and their own personal best position. While in SSO, spiders follow positions created by others' current positions and their previous positions, which positions are not guaranteed to visit the previous population, and different spiders can have different positions. Since the best global position and the current position of the spiders are different most of the time in the optimization process, the following two patterns cause different search behaviors. This may weaken the convergence ability of SSO, on the other hand, it can especially enhance the ability to solve multi-objective optimization problems with a large number of local optimization. ACO uses the path between the ant hive and food sources to introduce suitable solutions for the optimization problems. ACO was originally designed to solve combinatorial problems. Although, in recent years, variable ACO algorithms have also been designed to solve continuous problems [25].

The differences in the social spider optimization algorithm (SSO) lie in all aspects of the algorithm design[26]. The most important difference is that in SSO, spiders are classified by sex so male and female spiders have different search operations. But spiders in SSA search all the same. SSA also incorporates the information diffusion model into its algorithm design; Therefore, the social spider population in SSA is consistent with the IS model. Furthermore, SSA behavior mimics the foraging behavior of social spiders, while SSO mimics the mating behavior of social spiders. In SSO there are three spider movement operators executed first in parallel and then in sequence. The moving pattern of the third operator highly depends on the first two operators.

4-2- SSO: Initializing the population

The algorithm starts: by initializing the population S of N spider positions (solution).

The population contains females f_i and males m_i.

The number of females is randomly selected within the range of 65% - 90% and calculated by the following equation[27]:

N_f=floorp[(0.9-rand(0,1).0.25).N]

The number of male spiders Nm is calculated as follows.

Nm=N-Nf

The female spider position f_i is generated randomly between the lower initial parameter bound p^{low} and the upper initial parameter bound p^{high} as follows.

$$fi^0{,}j=p_j^{low}+rand(0,1).(p_j^{high}-p_j^{low})$$

$$i=1,2,\ldots,N_f\,;j=1,2,\ldots,n$$

The male spider position m_i is generated randomly as follows.

$$M_i^0{,}_j = p_j^{low} + rand(0,1).(p_j^{high}-p_j^{low})$$

$i=1,2,\ldots,N_m$;$j=1,2,\ldots,n$

4-3- SSO: Fitness evaluation

In the SSO algorithm, the weight of each spider represents the quality of the solution. The function value of each solution i is calculated as follow[23].

$$W_j = \frac{J(s_i) - woest_s}{best_s - worse_s}$$

Where $J(s_i)$ is the fitness value obtained of the spider position s_i, the values worst and bests are the maximum and the minimum values of the solution in the population respectively.

4-4- SSO: Vibrations through the communal web

The information among the colony members is transmitted through the communal web and encoded as small vibrations. The vibrations depend on the weight and distance of the spider which has generated them. The information transmitted (vibrations) perceived by the individual i from member j are modeled as follows [23].

$$Vibi_{,j} = w_j . e^{-d2}i,j$$

Where the d_{ij} is the Euclidian distance between the spiders i and j.

5- MODELING OF THE VIBRATIONS SSO THROUGH THE COMMUNAL WEB

The SSO search space is designed as a web, where each position is related to a possible solution to the problem and corresponds to the fitness value of the objective function. The quality solution (fitness) corresponds to the potential of finding a food source. As each spider moves to a new location, the other spiders notice the movement because the vibrations can be transmitted through the web, thus sharing collective social knowledge between them.

There are three relationships of the vibrations between any pair of spiders as follows.Vibrations$Vibc_i$. The transmitted information (vibrations) between the individual i and the member c (s_c), which is the nearest member to i with a higher weight is defined as follows.

$$Vibc_i = w_c . e^{-d2}i,c$$

Vibrations $Vibb_i$. The transmitted information (vibrations) between the individual i and the member b (s_b) which is the best member in the population S can be defined as follow.

$$Vibb_i = w_b . e^{-d2}i,b$$

Vibrations $Vibf_i$. The transmitted information (vibrations) between the individual i and the nearest female individual f(s_f) is defined as follows.

$$Vibf_i = w_f . e^{-d2}i,f$$

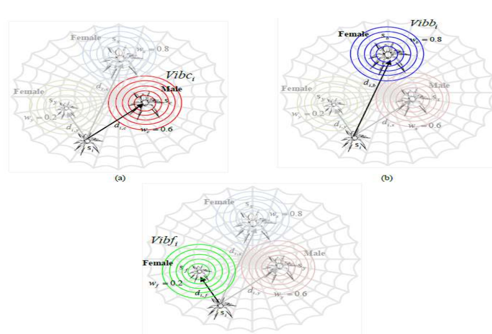

Fig-3: Configuration of each special relation: a)V_ibc_i, b)$V_ibb_i$and c)V_ibf_i [25]

Male cooperative

- D: The male spider with a weight value above the median value of the male population.
- ND: The other males with weights under the median.
- The dominant spider has better fitness and they are attracted to the closest female spider in the communal web.
- Non-dominant male spiders tend to concentrate in the center of the male population as a strategy to take advantage of resources wasted by dominant males.
- The position of the male spider:

$$M_i^{t+1}=\begin{cases} m_i^t+a.Vibf_i.(s_f-m_i^t)+\eth.(rand-0.5) & \text{if } w_{Nf+i}>w_{Nf+m} \\ m_i^t+a.\left(\frac{\sum_{h=1}^{Nm} m_h^t.w_{nf+h}}{\sum_{h=1}^{Nm} w_{Nf+h}}-m_i^t\right) \end{cases}$$

Where s_f represents the nearest female spider to the male spider i and W is the median weight indexed by N_{f+m} of the male spider population.

Female cooperative operator

- The movement of attraction or repulsion of a female spider i at time step t+1 is developed over spiders according to their vibrations
- Generate A uniform random number $r_m$within the range [0,1].
- If r_m is smaller than a threshold PF, it causes an attraction movement. Otherwise, it causes a repulsive movement:

$$F_i^{t+1}=\begin{cases} f_i^t+a.Vibc_i.(s_c-f_i^t)+\beta.Vibb_i.(s_b-f_i^t)+\eth.(rand-0.5) & \text{with probabilityPF} \\ f_i^t-a.Vibc_i.(s_c-f_i^t)-\beta.Vibb_i..(s_b-f_i^t)+\eth.(rand-0.5) & \text{with probability1-PF} \end{cases}$$

- Where r_m,α,β,δ, and *rand* are uniform random numbers between [0, 1], and $s_b$and$s_c$represent the nearest member to i that holds the best spider of the entire population and higher weight, respectively.

Mating operator

- The mating in a social spider colony: by the dominant males and the female members.
- Forming a new brood: placement of a set E_g of female members in a specific range r by a dominant male m_g spider,

$$r=\frac{\sum_{j=1}^{n}(p_j^{high}-p_j^{low})}{2.n}$$

- Where n is the dimension of the problem, and l_j^{high} and l_j^{low} are the upper and lower bounds.
- When the new spider is formed, it is compared to the worst spider in the colony. If the new

spider is better, the worst spider is replaced with the new spider.

According to the explanation given, ubiquitous computing works with thousands of thousands of systems around the world; that quality, efficiency, and reliability are the first words in these communications. On the other hand, optimization algorithms have shown their optimality in every field; Therefore, with the help and implementation of one of the best optimization algorithms called a social spider, we can have an effect on these communications and transmissions, which we do not claim to be of the highest quality; But it is possible to imitate the life of these spiders and improve the network as much as possible.

6- EVALUATION

6-1- Research data

Tavallaee et al. (2009) [28] presented a new dataset called NSL-KDD, which consists of selected files of KDD dataset and does not suffer from any shortage. In this research, like other valid articles in this field, the mentioned dataset is used.

6-2- Comparison

To evaluate the method presented in this research, several criteria should be considered. These criteria include:

- The feasibility of the implementation in the real world.
- Ensuring the optimal deployment of nodes.

To evaluate and compare the proposed method, the proposed work is compared with artificial fish, bees, and genetic algorithms[29].

To evaluate the performance of the proposed method, the method was compared with known algorithms. The simulation parameters are listed in the table below [30-35]. The following indicators are considered in the evaluation:

1) Network lifetime.
2) The number of remaining nodes over time.
3) Total remaining energy of nodes in each round.

Table-1: Simulation parameters

The number of Nodes is N	100, 200
Network size	$100 \times 100\ m2$
Primary energy	$E0 = 0.5J$
probability CH (P)	0.1
Energy cost of data collection	$EDA = 50nJ/bit$
closed size	4000 bits
Transmitter/Receiver Electronica	$Eelec = 50nJ/bit$

Transmission amplifier	$\epsilon fs = 10pj/bit/m2$
Transmission amplifier	$\epsilon mp = 0.0013pj/bit/m4$
$\boldsymbol{\alpha}$ **(super)**	1
$\boldsymbol{\beta}$ **(advanced)**	2
$\boldsymbol{m}$	0.5
$\boldsymbol{m0}$	0.4
Probability of fitness function $\boldsymbol{p1}$	0.7
Probability of fitness function$\boldsymbol{p2}$	0.3

Figure 4 shows the comparison of the number of live nodes against the number of repetitions (rounds) of 100 nodes;

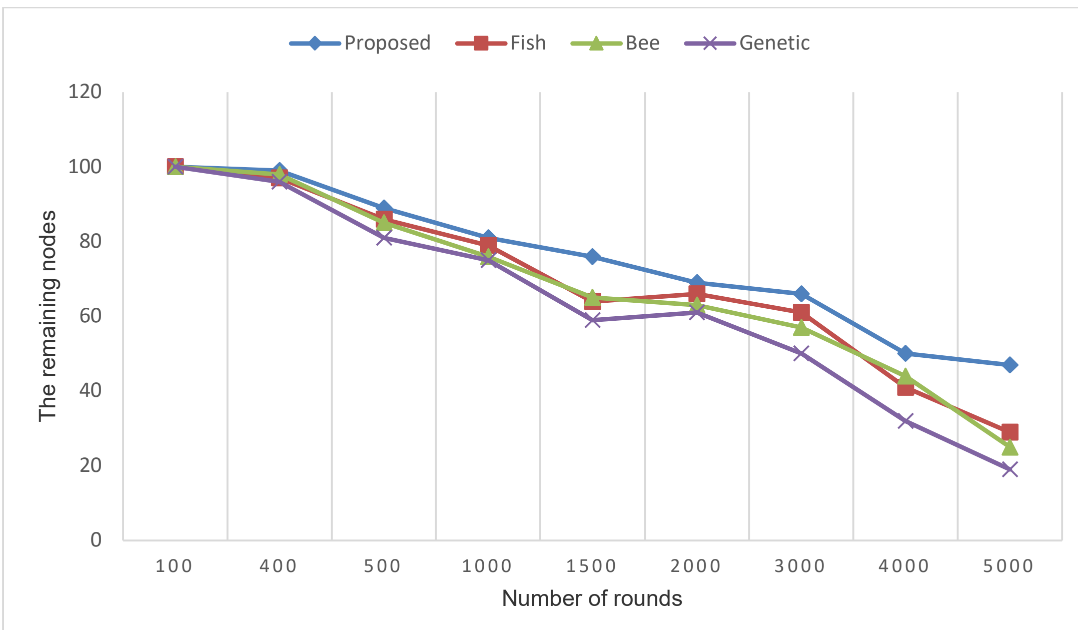

Fig-4:Network lifetime, $N = 100$

From the experiments, it can be concluded that the proposed method has more remaining nodes than other algorithms; So, with the help of the presented method, the lifetime of the network can be increased, because the error rate has decreased, and the efficiency and reliability have increased accordingly. Figure 5 analyzes the reliability of the network among the existing methods.

Figure 6 shows the performance of all considered algorithms.

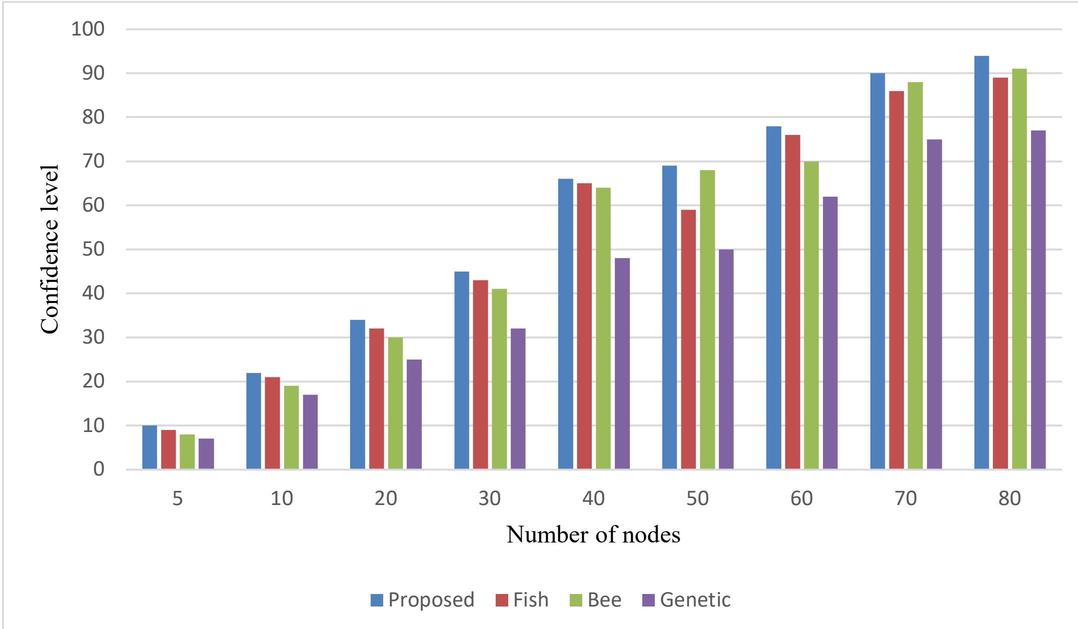

Fig-5:Comparison of reliability

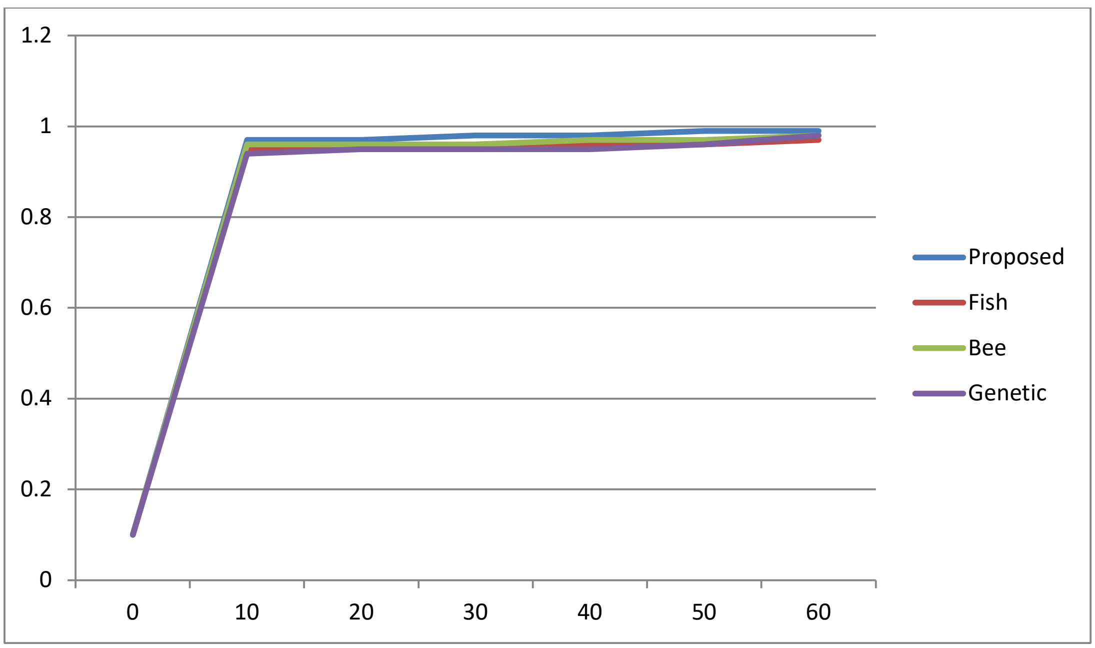

Fig-6:Ratio between time and throughput

The performance of the social spider optimization algorithm is slightly better than other algorithms. Therefore, according to the comparisons, the performance and speed of data transfer by the social spider optimization algorithm are better than the other algorithms under review.

7-CONCLUSION

Ubiquitous computing is a method of performing calculations in a vast and huge network like the Internet. Ubiquitous computing systems have been introduced to effectively use Internet-connected resources on a global scale.

In this article, we tried to first understand ubiquitous computing, examine its challenges and limitations, then look for a solution to improve it. Therefore, we decided to use the social spider optimization algorithm because optimization algorithms are a good solution to improve the quality of systems. The optimization algorithm of the social spider can be used to increase the efficiency and quality of communications and transmissions of ubiquitous computing systems.

The results of the comparisons show that the proposed method can be highly effective in increasing efficiency and improving network reliability. A comparative analysis between the social spider algorithm, bee, and artificial fish algorithm has been done based on parameters such as network length and throughput. The performance and data transfer speed of the social spider optimization algorithm are better than other algorithms considered in this research.

REFERENCES

[1] Yilmaz, E., Ecer, A.,Parallel CFD Applications Under DLB Environment, in Parallel Computational Fluid Dynamics 2001, 2002

[2] Matthew, N. O.,Sadiku, Kelechi, G. Eze, and Sarhan, M. Musa, GLOBAL COMPUTING, International Journal of Advances in Scientific Research and Engineering (ijasre), Volume 5, Issue 5 May - 2019

[3] Aida, K., et al., Performance evaluation model for scheduling in global computing systems, The International Journal of High Performance Computing Applications , vol. 14, no. 3, Fall 2000, pp. 268-279.

[4] Pfoser D.,Pitoura, E., andTryfona, N., Metadata modeling in a global computing environment, Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems, 2002, pp. 68-73.

[5] Takefus, A., et al., Multi-client LAN/WAN performance analysis of Ninf: a high-performance global computing system, Proceedings of the 1997 ACM/IEEE Conference on Supercomputing, San Jose, CA., Nov. 1997, pp. 1-23.

[6] Aleti, A., Moser, I., and Meedeniya, I., Choosing the appropriate forecasting model for predictive parameter control, Evolutionary Computation, vol. 22, no. 2, pp. 319–349, 2014.

[7] Smit, S., andEiben, A., Comparing parameter tuning methods for evolutionary algorithms, in Proc. IEEE Congress on Evolutionary Computation (CEC), Trondheim, Norway, May 2009, pp. 399–406.

[8] Grefenstette, J., Optimization of control parameters for genetic algorithms,IEEE Transactions on Systems, Man and Cybernetics, vol. 16, no. 1, pp. 122–128, 1986.

[9] Shi, Y., and Eberhart, R. C., Parameter selection in particle swarm optimization, in Proceedings of the 7th International Conference on Evolutionary Programming VII, 1998, pp. 591–600.

[10] Qin, A. K., and Li, X., Differential evolution on the CEC-2013 singleobjective continuous optimization testbed, in Proc. IEEE Congress on Evolutionary Computation (CEC), Cancun, Mexico, Jun. 2013, pp. 1099–1106.

[11] Yu, J. J. Q., and Li, V. O. K., A social spider algorithm for global optimization, Appl. Soft Comput., vol. 30, pp. 614–627, 2015.

[12] Uetz, G., Foraging strategies of spiders, Trends in Ecology and Evolution, vol. 7, no. 5, pp. 155–159, 1992.

[13] Yu, J. J. Q., and Li, V. O. K., Base station switching problem for green cellular networks with social spider algorithm, in Proc. IEEE Congress on Evolutionary Computation (CEC), Beijing, China, Jul. 2014, pp. 1–7.

[14] ——, "Power-controlled base station switching and cell zooming in green cellular networks, 2015, submitted for publication.

[15] Cuevas, E., Cienguegos, M., Zaldvar, D., Perez Cisneros, M.: A swarm optimization algorithm inspired in the behavior of the social-spider. Expert Systems with Applications 40(16), 6374–6384 (2013)

[16] James J.Q, Victor O.K.: A Social Spider Algorithm for Global Algorithm. Technical Report No. TR 2013-004, The University of Hong Kong, October 2013

[17] Cuevas, E., Cienfuegos, M.: A new algorithm inspired in the behavior of the socialspider for constrained optimization. Expert Systems with Applications 41(2), 412–425 (2014)

[18] Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of global optimization 39(3), 459–471 (2007)

[19] Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Applied soft computing 8(1), 687–697 (2008)

[20] Baker, M., Buyya, R., and Laforenza, D., The Grid: International efforts in global computing, https://pdfs.semanticscholar.org/333b/8725e7c7b06ed0d55ab64cfe792812ff687f.pdf

[21] Alexandrov, A. D. , et al., SuperWeb: Research issues in Java-based global computing, Concurrency: Practice and Experience, vol.9, no. 6, June 1997, pp. 535-553.

[22] Carde, L., Global computation, ACM Computing Surveys, vol. 28, no. 4, Dec, 1996, pp. 66-68.

[23] Cuevas, E., Cienfuegos, M., Zaldívar, D., Pérez-Cisneros, M. A swarm optimization

algorithm inspired in the behavior of the social-spider, Expert Systems with Applications, 40 (16), (2013), pp. 6374-6384

[24] http://www.slideshare.net/afar1111/social-spider-optimization

[25] Socha, K., Dorigo, M., Ant colony optimization for continuous domains, European Journal of Operational Research 185 (3) (2008) 1155{1173.

[26] Cuevas, E., M. Cienfuegos, D. Zaldvar, M. Prez-Cisneros, A swarm optimization algorithm inspired in the behavior of the social-spider, Expert Systems with Applications 40 (16) (2013) 6374C{6384.

[27] MetwalliAnter, A., Social-Spider Optimization Algorithm, Faculty of Computers and informatics, 6 June 2015, BeniSuef University, Member of the Scientific Research Group in Egypt, http://www.egyptscience.net

[28] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A. A., Jul. 2009. A detailed analysis of the KDD CUP 99 data set. In: Proc. 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6, https://doi.org/10.1109/CISDA.2009.5356528

[29] Liu, L., Yu, W., Liang, F., Griffith, D., Golmie, N., On deep reinforcement learning security for Industrial Internet of Things, Computer Communications, Available online 8 January 2021

[30] Kumar A., Jacold L., and Ananda A. L. (2004) "SCTP Vs TCP: Performance comparison in MANETs." Proceedings of 29th Annual IEEE International Conference on Local Computer Networks 1: 431 – 432.

[31] Kim D., Bae H., and Toh C.K. (2007) "Improving TCP-vegas performance over MANET Routing protocols." IEEE Transactions on Vehicular Technology 56(1): 372 – 377.

[32] Abouzeid A. A., and Azizoglu M. (2003) "Comprehensive performance analysis of a TCP session over a Wireless Fading Link with Queueing." IEEE Transactions on Wireless Communication 2(2):344 - 356.

[33] Manshahia M. S. (2016) "Wireless sensor networks: A survey.", International Journal of Scientific & Engineering Research 7(4): 710-716.

[34] Manshahia M. S. (2015) "Water wave optimization algorithm based congestion control and quality of service improvement in wireless sensor networks." Transactions on Networks and Communications 5(4): 31-39

[35] Manshahia M. S., Dave M., and Singh S.B. (2016) "Firefly algorithm based clustering technique for wireless sensor networks." Proc. International Conference on Wireless Communications, Signal Processing and Networking, Chennai, India.

[36] Nguyen, D.; Ding, M.; Pathirana, P.; Seneviratne, A.; Li, J.; Poor, V. Cooperative Task Offloading and Block Mining in Blockchain-based Edge Computing with Multi-agent Deep Reinforcement Learning. IEEE Trans. Mob. Comput. 2021, 22, 2021–2037.[CrossRef]

[37] C. You and K. Huang, "Multiuser resource allocation formobile-edge computation offloading," in 2016 IEEE GlobalCommunications Conference (GLOBECOM), Washington,DC, USA, 2016.

[38] T.Tran and D. Pompili, "Joint task offloading and resourceallocation for multi-server mobile-edge computing networks,"IEEE Transactions on Vehicular Technology, vol. 68, no. 1,pp. 856–868, 2019.

[39] J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, andP. Mohapatra, "Edge cloud offloading algorithms: issues,methods, and perspectives," ACM Computing Surveys,vol. 52, no. 1, 2020.

[40] D. Han, W. Chen, and Y. Fang, "Joint channel and queue

[41] aware scheduling for latency sensitive mobile edge computingwith power constraints," IEEETransactions on Wireless Communications, vol. 19, no. 6, pp. 3938–3951, 2020

[42] X. Chen, Y. Cai, L. Li, M. Zhao, B. Champagne, and L. Hanzo,"Energy-efficient resource allocation for latency-sensitivemobile edge computing," IEEE Transactions on VehicularTechnology, vol. 69, no. 2, pp. 2246–2262, 2020.